ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

BY
Fereydoun FARRAHI MOGHADDAM

CARBON-PROFIT-AWARE JOB SCHEDULING AND LOAD BALANCING IN GEOGRAPHICALLY DISTRIBUTED CLOUD FOR HPC AND WEB APPLICATIONS

MONTREAL, 16 JANUARY 2014

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Mr. Mohamed Cheriet, Thesis Director
Department of Automated Manufacturing Engineering, École de Technologie Supérieure

Mr. Jean-Marc Robert, Committee President
Department of Software and IT Engineering, École de Technologie Supérieure

Mr. David Wright, External Examiner
Telfer School of Management, University of Ottawa

Mr. Michel Kadoch, Examiner
Department of Electrical Engineering, École de Technologie Supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

ON 12 NOVEMBER 2013

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

To Brayden,

ACKNOWLEDGEMENTS

First and foremost, I want to thank my supervisor Prof. Mohamed Cheriet. It has been an honor to be his Ph.D. student on Green ICT. He has taught me how good and practical research is done. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive.

I would like to thank the board of examiners, Profs. Jean-Marc Robert, David Wright, and Michel Kadoch, for their time, interest, and helpful comments.

I must thank my brother Dr. Reza Farrahi Moghaddam who is a research associate in the Synchromedia laboratory. We worked together on various topics, and I very much appreciated his enthusiasm, intensity, and willingness.

I must also thank all members of the Synchromedia laboratory, all staff of the department of Automated manufacturing engineering, and all administrative and technical staff of École de technologie supérieure.

I would like to thank my loved ones, who have supported me throughout the entire process, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love and for your support.

Finally, I acknowledge the financial support of the Canadian Network for the Advancement of Research, Industry and Education (CANARIE), Natural Sciences and Engineering Research Council of Canada (NSERC), and the ÉTS scholarship program, which made this research possible.

CARBON-PROFIT-AWARE JOB SCHEDULING AND LOAD BALANCING IN GEOGRAPHICALLY DISTRIBUTED CLOUD FOR HPC AND WEB APPLICATIONS

Fereydoun FARRAHI MOGHADDAM

ABSTRACT

This thesis introduces two carbon-profit-aware control mechanisms that can be used to improve performance of job scheduling and load balancing in an interconnected system of geographically distributed data centers for HPC[1] and web applications. These control mechanisms consist of three primary components that perform: 1) measurement and modeling, 2) job planning, and 3) plan execution. The measurement and modeling component provide information on energy consumption and carbon footprint as well as utilization, weather, and pricing information. The job planning component uses this information to suggest the best arrangement of applications as a possible configuration to the plan execution component to perform it on the system.

For reporting and decision making purposes, some metrics need to be modeled based on directly measured inputs. There are two challenges in accurately modeling of these necessary metrics: 1) feature selection and 2) curve fitting (regression). First, to improve the accuracy of power consumption models of the underutilized servers, advanced fitting methodologies were used on the selected server features. The resulting model is then evaluated on real servers and is used as part of load balancing mechanism for web applications. We also provide an inclusive model for cooling system in data centers to optimize the power consumption of cooling system, which in turn is used by the planning component. Furthermore, we introduce another model to calculate the profit of the system based on the price of electricity, carbon tax, operational costs, sales tax, and corporation taxes. This model is used for optimized scheduling of HPC jobs.

For position allocation of web applications, a new heuristic algorithm is introduced for load balancing of virtual machines in a geographically distributed system in order to improve its carbon awareness. This new heuristic algorithm is based on genetic algorithm and is specifically tailored for optimization problems of interconnected system of distributed data centers. A simple version of this heuristic algorithm has been implemented in the GSN project,[2] as a carbon-aware controller.

Similarly, for scheduling of HPC jobs on servers, two new metrics are introduced: 1) profit-per-core-hour-GHz and 2) virtual carbon tax. In the HPC job scheduler, these new metrics are used to maximize profit and minimize the carbon footprint of the system, respectively. Once the application execution plan is determined, plan execution component will attempt to implement it on the system. Plan execution component immediately uses the hypervisors on

[1] Refer to Appendix I.1 for more details.

[2] Refer to Appendix III for more details.

physical servers to create, remove, and migrate virtual machines. It also executes and controls the HPC jobs or web applications on the virtual machines.

For validating systems designed using the proposed modeling and planning components, a simulation platform using real system data was developed, and new methodologies were compared with the state-of-the-art methods considering various scenarios. The experimental results show improvement in power modeling of servers, significant carbon reduction in load balancing of web applications, and significant profit-carbon improvement in HPC job scheduling.

Keywords: Carbon-Profit-Aware, HPC, Job Scheduling, Web Application, Load Balancing, Geographically Distributed Data Centers, Geographically Distributed Cloud, Carbon Tax, Virtual Carbon Tax, Multi-Level Grouping Genetic Algorithm, Server Power Metering, Cooling System Power Modeling, Profit-per-Core-Hour-GHz

ORDONNANCEMENT DE TÂCHES INFORMATIQUES ET RÉPARTITION DE CHARGE EN FONCTION DES PROFITS ET DES ÉMISSIONS DE CARBONE DANS DES NUAGES RÉPARTIS GÉOGRAPHIQUEMENT POUR LES APPLICATIONS HPC ET WEB

Fereydoun FARRAHI MOGHADDAM

RÉSUMÉ

Cette thèse présente deux mécanismes de contrôle en fonction des profits et des émissions de carbone, pour améliorer les performances d'ordonnancement de tâches et de répartition de charge, dans un système interconnecté de centres de données réparti géographiquement pour les applications HPC[3] et web. Ces mécanismes de contrôle sont constitués de trois composants primaires qui effectuent: 1) la mesure et la modélisation, 2) la planification de tâches, et 3) l'exécution du plan. La partie de mesure et modélisation fournissent des informations sur la consommation d'énergie et l'empreinte carbone ainsi que l'information concernant l'utilisation, coût, et de donnée météorologique. La partie de planification de tâches utilise ces informations pour proposer la meilleure disposition des applications à la partie d'exécution du plan, afin de l'exécuter sur le système.

Pour des fins de rapports et décision, certaines métriques doivent être modélisées en fonction de données mesurées directement. Il existe deux défis à la modélisation fidèle de ces métriques essentielles: 1) la sélection de caractéristiques et 2) l'ajustement des courbes (régression). Tout d'abord, afin d'améliorer la précision des modèles de consommation d'énergie des serveurs sous-utilisés, les méthodes d'ajustement des courbes avancées ont été utilisées sur les caractéristiques sélectionnées de serveur. Le modèle qui en résulte est ensuite évalué sur des serveurs réels et est utilisé par le mécanisme de répartition de charge pour les applications web. Nous fournissons également un modèle inclusif pour le système de refroidissement des centres de données afin d'optimiser sa consommation d'énergie, qui à son tour est utilisé par la partie de planification de tâches. De plus, nous introduisons un autre modèle pour calculer le bénéfice du système, basé sur le prix de l'électricité, taxe carbone, les coûts opérationnels, la taxe de vente et l'impôt des sociétés. Ce modèle est utilisé pour la planification optimisée des tâches HPC.

Pour l'allocation de position d'applications web, un nouvel algorithme heuristique est introduit pour la répartition de charge des machines virtuelles dans un système réparti géographiquement afin de diminuer l'empreinte carbone. Ce nouvel algorithme heuristique est basée sur un algorithme génétique spécialement conçu pour les problèmes d'optimisation de système interconnecté de centres de données répartie géographiquement. Une version simple de cet algorithme heuristique est mis en œuvre dans le projet GreenStar,[4] en tant que contrôleur de carbone.

[3] Voir l'annexe I.1 pour plus de détails.

[4] Voir l'annexe III pour plus de détails.

De même, pour l'ordonnancement des tâches HPC sur les serveurs, deux nouvelles métriques sont introduites: 1) Bénéfice-par-cœur-heure-GHz et 2) la taxe carbone virtuel. Dans l'ordonnanceur de tâches HPC, ces nouvelles métriques sont utilisées pour maximiser les profits et minimiser l'empreinte carbone du système. Une fois le plan d'exécution d'application est déterminé, la partie d'exécution du plan va tenter de mettre en œuvre le système. La partie d'exécution du plan utilise directement les hyperviseurs sur des serveurs physiques pour créer, supprimer, et migrer les machines virtuelles. Il exécute et contrôle également les tâches HPC ou des applications web sur les machines virtuelles. Pour valider le système conçu, utilisant la modélisation proposée et la planification de tâches, une plateforme de simulation utilisant les données du système réel a été développée, et nos méthodes originales ont été comparées avec les méthodes de la littérature, sous plusieurs scénarios différents. Les résultats expérimentaux montrent une amélioration dans la modélisation de la puissance des serveurs, une réduction importante de carbone lors de la répartition de charge des applications Web, et l'amélioration significative de profits et de carbone de l'ordonnancement de tâches HPC.

Mot-clés : Dépendance aux profits et émissions de carbone, HPC, Ordonnancement de tâches, Application web, répartition de charge, Centres de données répartie géographiquement, Nuage répartie géographiquement, Taxe carbone, Taxe carbone virtuelle, Algorithme génétique par regroupement multi-niveau, Modélisation de la puissance de serveurs, Modélisation de la puissance de système de refroidissement, Bénéfice-par-cœur-heure-GHz

CONTENTS

Page

INTRODUCTION 1
0.1 Context 1
0.2 Problem Statement 3
0.3 Objectives 7
0.4 Thesis Outline 9

CHAPTER 1 LITERATURE REVIEW 11
1.1 Network of Distributed Data Centers with Cloud Capabilities 11
1.1.1 Performance-Aware Scheduler 12
1.1.2 Energy-Aware Scheduler 15
1.1.3 Profit-Aware Scheduler 20
1.1.4 Other type of Schedulers 21
1.2 Server Consolidation and Load Balancing in Cloud Computing 24
1.2.1 Grouping Genetic Algorithm in Server Consolidation 27
1.2.2 Grouping Mechanism in Grouping Genetic Algorithm 28
1.3 Server Energy Metering 32
1.4 Cooling System Power Modeling 34
1.4.1 Computer room (CR) 37
1.4.2 Chillers 38
1.4.3 Cooling tower (CT) 40
1.4.4 Heat Handling Capacity in a Datacenter 41
1.5 Simulation Platforms for Energy Efficeiny and GhG Footprint in Cloud Computing 45
1.5.1 CloudSim 45
1.5.2 GreenCloud 46
1.5.3 iCanCloud 46
1.5.4 MDCSim 47
1.6 Chapter Summary 48

CHAPTER 2 CARBON-PROFIT-AWARE GEO-DISTRIBUTED CLOUD 49
2.1 State-of-the-Art Geo-DisC Architecture (Baseline Design) 49
2.1.1 Energy Model 52
2.1.2 Carbon Footprint and Pricing 53
2.1.3 Scheduler Features 54
2.1.4 HPC Workload Features 55
2.1.5 Summary 56
2.2 Carbon-Profit-Aware Geo-DisC Architecture (Our Proposed Design) 56
2.2.1 Component Modeling 58
2.2.2 Carbon-Profit-Aware Scheduler 59
2.2.3 MLGGA Load Balancer for Web Applications 59
2.2.4 Managers and Controllers 60

2.2.5 Summary 62
2.3 Chapter Summary 62

CHAPTER 3 GEOGRAPHICALLY DISTRIBUTED CLOUD MODELING 63
3.1 IT Equipment Modeling 63
3.1.1 Profit per Core-Hour-GHz 64
3.1.2 Power Metering Model for Servers 66
3.1.3 NDC Carbon-Related Metrics 68
3.2 Cooling System Modeling 69
3.2.1 The Temperature Altitude Aware Model (TAAM) 69
3.2.2 Set of Equations of the Cooling System Model 71
3.2.3 Summary 74
3.3 Chapter Summary 74

CHAPTER 4 CARBON-PROFIT-AWARE JOB SCHEDULER 75
4.1 Scheduling Metrics 75
4.1.1 Energy and Carbon 76
4.1.2 Carbon Tax 76
4.1.3 Profit per Core-Hour-GHz 76
4.1.4 Summary 78
4.2 Optimization Problem 79
4.3 CPA Scheduler Algorithm 80
4.3.1 Optimum Frequency Calculation 82
4.3.2 Virtual Carbon Tax 87
4.3.3 Summary 90
4.4 Expected Outcome 91
4.4.1 Performance 91
4.4.2 Virtual Carbon Tax 92
4.5 Chapter Summary 92

CHAPTER 5 CARBON-AWARE LOAD BALANCER 93
5.1 Multi-Level Grouping Genetic Algorithm 93
5.1.1 MLGGA Crossover 94
5.1.2 MLGGA Mutation 97
5.1.3 Extensions of the MLGGA Crossover and Mutation 98
5.2 Carbon-Aware Load Balancing Concept 100
5.3 Chapter Summary 103

CHAPTER 6 EXPERIMENTAL RESULTS AND VALIDATION 105
6.1 Simulation Environment 105
6.1.1 Batch Simulation 105
6.1.2 Caching 106
6.1.3 Summary 107
6.2 Green HPC Job Scheduling Scenarios 107
6.2.1 Experimental Setup 107

6.2.1.1 Comparing Algorithms 111
6.2.2 CPA Scheduler Performance Study 112
6.2.3 Seasonal Energy-Variations Study 121
6.2.4 Cooling System Study 123
6.2.5 Virtual Carbon Tax Study 125
6.2.5.1 Carbon-Profit Trade-Off in CPAS with VCT 128
6.2.5.2 Study of CPA Scheduler based on Virtual GHG-INT Equivalent Carbon Tax 131
6.2.6 Summary 133
6.3 Server Power Metering Validation 134
6.3.1 Experimental Setup 135
6.3.1.1 Server Power Metering Setup 136
6.3.1.2 VM migration Power Metering Setup 136
6.3.2 Server Power Metering Validation Results 136
6.3.3 VM Migration Power Metering Validation Results 138
6.4 Low-Carbon Web Application Load Balancing 139
6.4.1 Experimental Setup 140
6.4.1.1 Optimization Problem 141
6.4.2 MLGGA Performance Analysis on Large Scale CADCloud 143
6.4.2.1 MLGGA Comparison Results 143
6.4.3 Energy Diversity Study 145
6.4.3.1 Results 146
6.4.4 MLGGA Performance Study on Real Data 150
6.4.5 MLGGA Convergence Time 150

CONCLUSION 151

ANNEX I DEFINITIONS 159

ANNEX II A MODIFIED GHG INTENSITY INDICATOR: TOWARD A SUSTAINABLE GLOBAL ECONOMY BASED ON A CARBON BORDER TAX AND EMISSIONS TRADING 163

ANNEX III GREENSTAR NETWORK PROJECT 173

BIBLIOGRAPHY 175

LIST OF TABLES

Page

Table 1.1 The comparison table among state-of-the-art approaches. 24

Table 1.2 Comparison of cloud computing simulators 47

Table 4.1 A color code describing the status of the scheduled jobs 86

Table 6.1 Various archives and source of real traces of HPC jobs. 108

Table 6.2 Some of the most cited HPC traced in the literature. 108

Table 6.3 Energy mix data 109

Table 6.4 Energy price data 109

Table 6.5 Energy mix and price data sources 109

Table 6.6 Carbon tax rates. 110

Table 6.7 The comparison table among schedulers used in experimental setup. 112

Table 6.8 The comparison table for performance study 120

Table 6.9 The comparison table for seasonal study of PERF algorithm 122

Table 6.10 The comparison table for seasonal study of CPAS algorithm 123

Table 6.11 The comparison table for cooling study 125

Table 6.12 The comparison table for virtual carbon tax study 127

Table 6.13 The comparison table for virtual carbon tax study of CPAS algorithm 130

Table 6.14 MLGGA performance study: 24-hour carbon and energy measurements 145

Table 6.15 MLGGA performance study: 24-hour carbon measurement with weather change 146

Table 6.16 Energy diversity study: ration of sources of energy in different scenarios 147

LIST OF FIGURES

Page

Figure 0.1 Sample schedule of HPC jobs 6

Figure 0.2 Research focus areas of this thesis 10

Figure 1.1 GGA representation for parent chromosomes. 30

Figure 1.2 GGA crossover in progress. 31

Figure 1.3 GGA crossover final result. 31

Figure 1.4 The chiller plant (cooling system) overview. 37

Figure 2.1 Geo-DisC baseline schema 50

Figure 2.2 Carbon-Profit-Aware Geo-DisC schema 57

Figure 2.3 Carbon-profit-aware Geo-DisC stacked graph 60

Figure 2.4 Carbon-profit-aware Geo-DisC control cycle 61

Figure 4.1 Geo-DisC maximum profit per core-hour 78

Figure 4.2 Profit per core-hour color code 78

Figure 4.3 Profit per CPU frequency graph 83

Figure 4.4 Color code of scheduled jobs 85

Figure 4.5 Optimum frequency for maximum profit 87

Figure 4.6 Optimum CPU frequency with sale rate = 2¢ per core-hour 88

Figure 4.7 Optimum CPU frequency with sale rate = 4¢ per core-hour 89

Figure 4.8 Optimum CPU frequency with sale rate = 6¢ per core-hour 90

Figure 4.9 Cost breakdown of a typical system 91

Figure 5.1 MLGGA representation for parent chromosomes. 97

Figure 5.2 MLGGA crossover in progress. 98

Figure 5.3 MLGGA crossover final result. 99

Figure 5.4 CADCloud Schema. 101

Figure 6.1 HPC workload features 108

Figure 6.2 Data centers greenness, electricity price rates, and environment temperature 111

Figure 6.3 Geo-DisC profit 113

Figure 6.4 Geo-DisC energy consumption 114

Figure 6.5 Geo-DisC average PUE 114

Figure 6.6 Geo-DisC carbon footprint 115

Figure 6.7 Geo-DisC greenness 115

Figure 6.8 Geo-DisC average frequency 116

Figure 6.9 Scheduled jobs plot by PERF algorithm 117

Figure 6.10 Scheduled jobs plot by CARB algorithm 117

Figure 6.11 Scheduled jobs plot by CPAS algorithm 118

Figure 6.12 Optimum profit map 118

Figure 6.13 Profit per frequency for CPAS algorithms 119

Figure 6.14 Profit per frequency for different algorithms 120

Figure 6.15 Sankey diagram of the cost and profit of the system 121

Figure 6.16 Geo-DisC in different seasons under PERF algorithm 122

Figure 6.17 Geo-DisC energy consumption in different seasons under CPAS algorithm 123

Figure 6.18 Geo-DisC carbon footprint (cooling system study) 124

Figure 6.19 Geo-DisC under CPAS algorithm with different cooling strategies 125

Figure 6.20 Geo-DisC under different algorithm with utilization of virtual carbon tax 126

Figure 6.21 Geo-DisC sale, costs, and profit for high virtual-carbon-tax scenario and scheduled by CPAS 127

Figure 6.22 Sankey diagram of the cost and profit of the system with VCT 128

Figure 6.23 Geo-DisC under CPAS algorithm with different virtual carbon taxes.....129

Figure 6.24 Geo-DisC sale, costs, and profit (scheduled by CPAS) 130

Figure 6.25 Carbon-profit trade-off per hour (ct represent the amount of VCT applied) 131

Figure 6.26 Geo-DisC under CPAS algorithm with different virtual "GHG indicator" taxes 134

Figure 6.27 Power prediction model 138

Figure 6.28 VM migration power prediction.....139

Figure 6.29 Distributed cloud in 11 cities. 142

Figure 6.30 Comparison of various methods with respect to carbon footprint. (50% load) 144

Figure 6.31 Comparison of various methods with respect to energy consumption. (50% load).....144

Figure 6.32 CADCloud Carbon measurement 147

Figure 6.33 CADCloud Sun sensitivity 148

Figure 6.34 CADCloud wind sensitivity 149

Figure 6.35 The G factors of various scenarios 149

Figure 6.36 Geo-DisC under load balancing algorithms 150

LIST OF ABBREVIATIONS

CADCloud	Carbon-Aware Distributed Cloud
CDU	Cabinet power Distribution Unit
CPA	Carbon-Profit-Aware
CPAS	Carbon-Profit-Aware Scheduler
CPU	Central Processing Unit
CRAC	Computer Room Air Conditioner
CT	Cooling Tower
DVFS	Dynamic Voltage and Frequency Scaling
FFD	First Fit Decreasing
Geo-DisC	Geographically Distributed Cloud
GGA	Grouping Genetic Algorithm
GhG	Greenhouse Gases
GSN	GreenStar Network
HPC	High Performance Computing
ICT	Information and Communications Technology
LCA	Life Cycle Assessment
LLC	Last Level Cache
MIP	Mixed Integer Program
MLGGA	Multi-Level Grouping Genetic Algorithm
MLGGA-CA	Carbon Aware MLGGA

MLGGA-EA	Energy Aware MLGGA
NDC	Network of Data Centers
NGO	Nongovernmental Organizations
NO-CONS	No Consolidation
PDU	Power Distribution Unit
PLR	Piecewise-Linear Regression
PMC	Performance Monitoring Counter
PpCHG	Profit-per-Core-Hour-GHz
PUE	Power Usage Effectiveness
VCT	Virtual Carbon Tax
WAN	Wide Area Network

INTRODUCTION

This research mainly deals with environmental impacts of geographically distributed data centers. In the rest of this chapter the context of known problems regarding this topic are firstly presented. Next, in order to address those problems or improve their currently available solutions, the objectives of this research are defined. Last, the outline of the research is presented.

0.1 Context

With the introduction of semiconductor, transistor, and integrated circuit technologies in mid-twentieth century, a new phase of achievements has started in the history of humankind, which rapidly improved his quality and style of life. Laptops, Internet, and smart phones are good examples of such improvements. All these new technologies put us in the middle of a new age of information and communications technology (ICT), which is changing the whole dynamic of social life, economy, and even politics. Accessing data is becoming a daily need for people as well as many sectors of industry. The new ICT technologies are usually based on data transfer and information processing, which highly depend on data centers. With more need for new ICT technologies, more data centers are required, which consequently causes more energy consumption in this sector.

However, this is not the whole story. There are known negative impacts and wastes related to any of these new technologies and any kind of energy production such as greenhouse gases (GhG) emissions. These negative impacts and wastes are destroying our ecosystem with the same speed as new technologies are improving our style and quality of life. Earth surface per capita is only 0.07 Km^2 right now, and it is decreasing. There is no far and safe place in the earth to release the wastes without negative impacts on our life. The relevant question here is that how long the oceans, the atmosphere and the land can sustain these adverse impacts.

Global warming and its impacts on our life are among the twenty-first century's biggest challenges for human societies. Greenhouse gases (GhG), especially carbon emissions, are the main man-made contributors to the global warming, an issue that is becoming a major concern

for many governments and NGOs. Although CO_2 emission is the main concern to be addressed in this research, but it is only one of the items in the long list of environment impacts of manufacturing and energy consumption of IT equipment. Considering possible correlation between CO_2 emissions and some of the other environmental impacts (Laurent *et al.*, 2012), by decreasing the CO_2 emissions, this research could indirectly contribute to a more environment-friendly solution with less overall environment impacts. For a detailed measure of impacts, a full Life Cycle Assessment (LCA) analysis is needed to be done which is out of scope of this research.

There is a trade-off between profit and environmental impacts. In the lack of proper environmental regulations, the current final cost of a product is not showing its real cost, therefore profit-profit-profit objective of many corporations does not consider the environmental impacts thoroughly. ICT sector is only one of contributors of CO_2 emissions, but this sector is growing fast. The contribution of the ICT sector currently represents no more than 2% of global GhG emissions (GeSI, 2008; McKinsey, 2007). However, considering the ICT enabling effect (GeSI, 2008), which pushes to increase the use of ICT and smart solutions to reduce the emissions of other sectors, the current rapid growth of ICT is expected to accelerate in the coming decades. This means that this sector will face an enormous challenge to reduce its own GhG emissions, which are a direct consequence of its higher energy consumption. Other contributing parameters are population growth, increase in percentage of the population who have access to ICT solutions, and shift of needs towards ICT solutions such as using Facebook which was not a daily need a decade ago. To address this issue, some states have already placed some regulations for carbon footprint, but many states do not have any regulations. A clear and accurate provisioning of carbon footprint in different granularities will help governments and also public to see and understand the scale of impact, and define the responsibility share of service providers and consumers in this important topic. Then, practical, fair and efficient regulation can be put in place.

Considering all above mentioned concerns, it is important for the ICT sector to improve its solutions to be more environment-friendly. It is not always easy to create environment-friendly solutions because of the trade off between higher performance and, for example, lower energy

consumption. Therefore, it is necessary that accurate and smart solutions be implemented in the major applications of ICT sector, especially in the hot spots of their power consumption, i.e. data centers. However, regarding this topic, not all type of ICT applications can be approached in a similar way because of their different characteristics and requirements. For example, web applications such as web services are usually run for a long time and the CPU demand of this type of application is variable. On the other hand, High Performance Computing (HPC) type of jobs are highly CPU demanding and the life time of these type of applications are short. Web applications may need high speed network connections, but this is not the case in most HPC jobs. This is the reason that in this research the focus for HPC jobs is on the initial scheduling, because it is unlikely that the HPC jobs, which have short life time, are moved to another location after initial placement. In contrast, the focus for web applications are on load balancing. There are other type of ICT applications such as telecommunication class of applications in which constraints are on the quality of service of calls, such as maximum acceptable response time to a call request. This class is out of scope of this research. Based on these characteristics and requirements, the solution need to be specifically designed and adapted to fulfill the main objectives of the system.

0.2 Problem Statement

As it was indicated in the above discussion, CO_2 emission is one of the main concerns of humankind in this century. There are many solutions which already proposed to address this issue in the ICT sector. Here, performance and coverage of these solutions with respect to their objectives and also type of applications will be discussed.

For an ICT service provider, the common practice for energy efficiency is to move their services from physical servers to virtual servers (virtual machines) as a server consolidation strategy. In theory, the services can be run on completely isolated containers with only a small increase in overhead processing caused by the hypervisors. There are security and reliability concerns related to this strategy which need to be thoroughly addressed, but is out of scope of this research. By server consolidation, a smaller number of servers with higher computing capabilities replace a larger number of underutilized servers. The direct result of this strategy is a

reduction in power consumption, which might lead to less GhG emissions. Even though energy efficiency is usually associated with less cost and less carbon footprint, but loss of profit, energy consumption, and carbon footprint are not necessary correlated in all the time. Therefore, less energy consumption is not always the best possible case for the businesses.

There may be different reasons for businesses to have several data centers in geographically distributed regions such as best pricing, resource management, hypervisor benefits, geographical advantages, redundancy, and security (Hwang *et al.*, 2013). These distributed data centers can be used to execute different type of tasks and services. Comparing two data centers, first, from a profit prospect, if the energy price in one data center is much less than the other one, it maybe more profitable to run services on the first data center than the other data center with higher energy price even if the power consumption in the first data center is a little bit more. Second, from a carbon footprint point of view, it makes sense to run the services on a data center with lower carbon emission rate than other data centers with higher carbon emission rates, even if the power consumption is higher in this data center compare to the other ones. To be able to answer accurately to this question that which data center is more suitable for running the services, one must consider all the parameter of the system such as type of service, price of energy, energy mix, profit, environment temperature, and cooling system.

There are three major phases in these kind of scenarios: data collection and modeling, planning, and execution. Modeling is an important part in the whole system because of three points: 1) measurement devices are not necessary cheap investment, and besides that they need installation and continuous maintenance, 2) some components of the system are not reachable for direct measurements such as memory power consumption or CPU power consumption, and 3) It is not possible to measure a component's future state. The modeling can have an estimated answer for all of these situations. Next phase is the planning, which is responsible for deciding for jobs and services execution time and place. Finally, a component is needed in the system to execute the generated plan.

To have a better result in the system, more accurate and complete models should be developed and used for the almost all the important components and sub components of the system. There

are a few state-of-the-art models for energy metering of a physical server, but they can be improved for more accuracy.

Cooling play a big role in the power consumption of a data center. Therefore, having an accurate model for its power consumption is essential. Because of the high number of components in a complete cooling system, the complexity of its models are very high, as well. Therefore, there are not many researches in the literature to consider all the parameters of a complete cooling system. Another parameter, which is important for businesses active in this area, is the modeling of the possible profit. Because there are many parameters which they affect the profit of such system, up to our best knowledge, there is no research done which is considering most of these parameters such as energy price, carbon tax, and sales and corporation tax. It is worth noting that, in some states and provinces, instead of the carbon tax, carbon credit is used in a carbon credit exchange market or emission trading system in order to control the carbon footprint of businesses. Carbon credit is out of scope of this research.

Having proper models to measure and estimate different metrics of the systems, different scenarios of network of interconnected geographically distributed data centers can be optimized based on the goals of the system. Goals of the system can be profit, energy efficiency, carbon footprint reduction, quality of service, or a combination of any of these goals. However, as it was mentioned in the context discussion, the solutions vary from one type to another type of applications. In this research we will discuss two main type of applications: HPC jobs and web applications.

For HPC jobs, one scenario is to have several data centers in different regions, and use a scheduler to choose the best data center for coming jobs to ensure the maximum profit. The simplest schedulers for this type of solution are greedy ones (for example Min-Min completion time (Braun *et al.*, 2001)). Figure 0.1 illustrate a sample job schedule for a few HPC jobs with different length (expected time to finish) and height (number of needed CPU cores).

There are some strategies for energy efficiency and carbon footprint reduction such as server consolidation and Dynamic Voltage and Frequency Scaling (DVFS) (Zhang *et al.*, 2010).

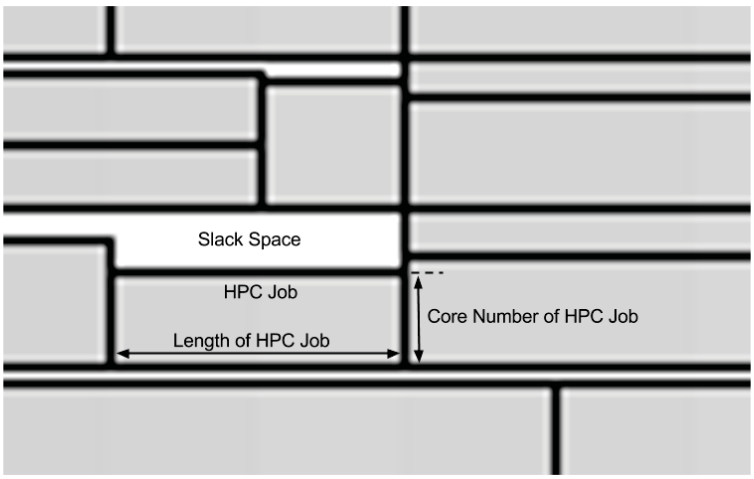


Figure 0.1 Sample schedule of HPC jobs

Server consolidation solution seeks for scheduling and moving the jobs and pack them on fewer servers. Therefore, other servers could be turned off or put in standby mode for energy efficiency and subsequently for carbon emission reduction. DVFS mainly seek for reducing the frequency of CPUs in the gaps between scheduled jobs or bringing the CPU frequencies to a pre-calculated optimum frequency for minimum energy consumption, and selecting the host servers based on their carbon emissions or profit (Garg *et al.*, 2011).

Nevertheless none of the above strategies can individually guarantee a very good performance under different circumstances of the system. Each strategy may have advantages in some circumstances. For example, if the job trace load is not 100%, server consolidation can be a solution. Yet, it is not logical to invest in an under-used system, unless this under utilization is based on accurate calculations and based on trade-off between performance of the system and its energy consumption. Also, when the strategy is to reduce the frequency for energy efficiency, if the CPU frequency is preoptimized and the load of the system is 100%, then the number of executed jobs will be less than when the frequency is maximum, therefore again the

system is underutilized and the performance of the system is compromised. In carbon-aware systems, the strategy of selective servers is useless when the load of the system is 100%. No matter which server is picked first, there are other jobs, which need to be placed on the other remaining servers.

For web applications, one scenario is having different data centers powered by different power mix. The data centers host some web applications and web applications are able to migrate between these data centers to achieve carbon footprint reduction or energy efficiency. Traditional server consolidation cannot be used on this type of systems because it simply does not consider the different energy profiles in each data center. Many state-of-the-art work are only for local data centers, and there is no heuristic method designed for distributed systems.

In order to test the new algorithms and schedulers, real systems would be required, but it is often too costly and unavailable. Therefore, the existence of a good simulation environment is vital for validating such system. The problem with simulation platform is that, for each newly proposed algorithm, there are new metrics and measures which may not already exist in previously used simulation platforms. For this reason, those metrics and algorithms need to be added to the existing simulation platform. Occasionally, a new simulation environment that is built from scratch makes more sense than modifying an existing simulator if the number of metrics is high. The other problem with the current simulation environments is their inability to loop through any given number of parameters within a range for comparison purposes. It is often necessary to execute multiple scenarios with different values of given parameters. Furthermore, each execution needs to be repeated until a proper result is obtained. In some cases, it is extremely time-consuming to execute these simulations one by one and execute them individually several times.

0.3 Objectives

The main objectives of this research are defined as follows:

- Obj #1: Designing a network of data centers system for HPC jobs and web applications with profit and environmental impact awareness.

  As mentioned in the problem statement (Section 0.2), there are a number of strategies which are adopted for job scheduling in a network of data centers environment such as performance-aware, energy-aware, profit-aware, and QoS-aware type of strategies. Each of them has its own metrics to measure and algorithms to schedule the jobs in the best position based on the defined objectives, separately. A performance-aware algorithm may maximize the amount of executed workload, but while the complexity of the system increases, there is no guarantee that doing so maximize the profit of the system. There is a similar argument about other strategies. In this research, one of main objectives is to consider most of these objectives together and create a comprehensive algorithm to find the best solution with the best results in terms of maximizing total profit and minimizing total environmental impacts with a clear control on their trade-off.

- Obj #2: Improving cooling system and server modeling.

  Modeling of servers is especially important when there is no measurement device present. In practice, not all the devices are connected to a measuring PDU. Even if they are, the modeling is still important for predicting the energy consumption of future situations or subsystems. For services, there is no measurement device and therefore predicting models are unavoidable. These models should also be accurate because of possible usage in financial calculation. Finally, they are necessary for simulation purposes when the system does not exist yet. It is also very important to see if the new models work correctly on real systems in small size.

- Obj #3: Designing a new scheduler to maximize profit and minimize the environmental impacts of an Network of Data Centers (NDC) simultaneously.

  As mentioned in the previous objectives, in the new design, all of the parameters are important and needed to be considered in the scheduler in order to have an efficient and realistic scheduler. In order to do so, these guidelines are adopted: i) Calculation of the amount of profit based on the best available information instead of assumption of corre-

lation between performance and profit, ii) Consideration of the current existing carbon regulations for carbon footprint reduction and introduction of new tools for intensifying the carbon emission reduction in absent of sufficient regulations. As mentioned earlier, not all states have a carbon policy to force businesses to reduce their carbon footprints. Therefore, in this research, one of objectives is to introduce an entity that acts as an intermediate factor to force the schedulers to consider more carbon reduction without major changes in the structure and objectives of the scheduler.

- Obj #4: Introduction of new heuristic algorithms for load balancing and consolidation.

  Distributed structures can give good benefits by adding diversity and choice to the system. Therefore, NDCs have new potential capabilities for achieving objectives of the system, but the complexity and topology of the systems can be also highly variable. In this research, one of objectives is to present a new heuristic algorithm tailored specially for NDCs consolidation problem.

- Obj #5: Developing a Simulation Platform.

  It is important to have simulation platform where energy, carbon, cost, and QoS can be integrated all together. Each structure may have different outcome under different circumstances. Therefore, it is important to test the new structures under various conditions such as diverse type of energy sources, workload, and system size.

0.4 Thesis Outline

The rest of the thesis is organized in 7 chapters. First, a complete literature review on state-of-the-art researches is provided in the Chapter 1. In Chapter 2, a general view of current network of data centers and a newly proposed system are presented. New models are introduced in Chapter 3 for energy metering of servers and cooling systems. In addition, a new model is also introduced for calculation of profit of a data center. Then, in the next chapter (Chapter 4), the main idea and mechanisms of the proposed scheduler are presented. In Chapter 5, a new genetic algorithm is introduced for load balancing and data center consolidation. Next, the experimental results and validations are reported in the Chapter 6. In this chapter, the essence

of the simulation platforms used in this research is also described. Last, a general conclusion is presented which will summarized the achievements of this research. The focus of research in this thesis is presented in Figure 0.2.

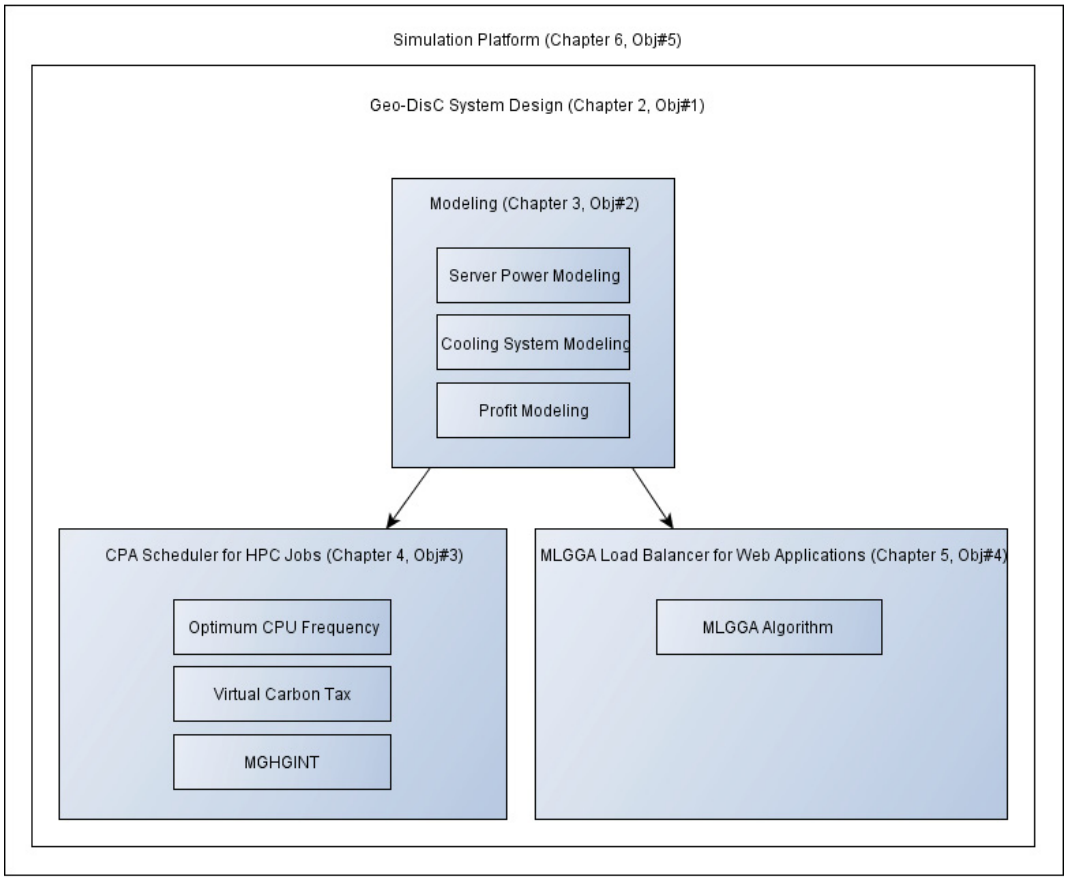


Figure 0.2 Research focus areas of this thesis

CHAPTER 1

LITERATURE REVIEW

As mentioned in the introduction, for different type of applications such as HPC and web, there are open research problems in data center measurement, modeling and optimization. In order to design a more efficient system to address these issues or improve their current solutions, existing state-of-the-art systems, methods, and models need to be discussed thoroughly.

In this chapter, first, currently implemented geographically-distributed-cloud structures and several type of job scheduler are presented in the Section 1.1. Next, server consolidation and load balancing solutions are presented in Section 1.2. Then, discussions on energy modeling are provided in Sections 1.3 and 1.4, and finally simulation platforms associated with network of data centers are discussed in Section 1.5.

1.1 Network of Distributed Data Centers with Cloud Capabilities

Network of distributed data centers (NDC) with cloud capabilities refers to several data centers which are positioned in geographically diverse locations and support compatible virtualization technologies. These data centers are well connected, and live VM migration is possible among them. Generally, a global scheduler is responsible to dispatch and manage the applications on these data centers. The type of this scheduler usually defines the type of the NDC. Therefore, the operation of an NDC can be imagined as a job scheduling problem. However, scheduling of jobs on computing resources cannot be always expressed as a single and unique problem statement. This is mainly because of high level of diversity in the possible compute configurations and also types, structures, and goals of computing jobs. That means that there is a spectrum of concepts that we may encounter when analyzing a specific configuration. In (Xhafa and Abraham, 2010), some of those concepts are described and discussed, and we just list them to show some of complexity of subject matter: heterogeneity of resource, heterogeneity of jobs, local schedulers, meta-scheduler, batch mode scheduling, resource-oriented schedulers, application-oriented schedulers, heuristic and metaheuristic methods for scheduling, local poli-

cies for resource sharing, job-resource requirements, and security. In particular, schedulers can be divided in three main categories: i) Local/Host Schedulers: The scheduler has the complete picture of its resources (free time slots, etc). In the case of super scheduler, see below, local schedulers also exist but they only follow the reserved slots determined by the super scheduler, ii) Meta Schedulers (Brokers): A meta scheduler distributes incoming jobs among a few local schedulers. Therefore, meta scheduler does not determine of the actual time slot assigned to a job. Instead, it uses the statistics of free resources reported by local schedulers to distribute the jobs, and iii) Super Schedulers: A super scheduler merges both local and meta scheduling strategies. Local schedulers first report actual free time slots to the super scheduler, and it then assigns jobs to them. This is the most efficient scheduler. However, it requires constant communication between schedulers, and solving the assignment problem could be very time consuming and ineffective especially in the case of high number of resources. In the following sections, several actual schedulers with goal functions toward performance, energy, carbon, profit, and QoS targets are presented.

1.1.1 Performance-Aware Scheduler

Scheduling of jobs on a distributed computing system, such as a network of data centers, is an old and well studied problem. For example, in Braun *et al.* (2001), several heuristic scheduling algorithms based on the makespan matrix, such as Opportunistic Load Balancing (OLB) and Minimum Completion Time (MCT), were introduced. It is worth noting that these algorithms consider assignment of jobs to machines, not to the processors. Therefore, their approach should be considered as a global scheduler. A global scheduler is not necessarily always a distributed scheduler. For example, a scheduler that works within a data center can work in a global manner, while it obviously is not a distributed scheduler. They also considered GA as one of their schedulers. For simulations, Expected Time to Compute (ETC) matrix were used. Also, proper synthesizd ETC matrices were used in order to simulate a heterogeneous computing environment.

Another milestone in the scheduling of computing resources is Freund *et al.* (1998), which introduced MaxMin and MinMin heuristic schedulers. Similar to Braun *et al.* (2001), only

performance in terms of completion time was considered, and there was no account for the energy consumption or carbon footprint of the operations. They performed their calculations using a simulated environment, called SmartNet.

In Maheswaran *et al.* (1999), the k-Percent Best (KBP) and Switching Algorithm (SA) schedulers were introduced along with the Minimum Execution Time (MET), Minimum Completion Time (MCT), Suffrage, and Opportunistic Load Balancing (OLB) heuristic schedulers. The k-percent best (KPB) scheduler considers only a subset of machines while mapping a job. The subset is formed by picking the $(k/100)m$ best machines based on their execution time of that job, where $100/m \leq k \leq 100$ and m is the number of machines. The job is assigned to a machine that provides the earliest completion time in the subset. The main idea behind KBP is not to map a job on the best machine, but it is to avoid mapping a job on a machine that could be a better choice for a yet-to-arrive job. If $k = 100$, then the KPB heuristic is actually reduced to the MCT heuristic. For the case $k = 100/m$, the KPB heuristic is equivalent to the MET heuristic.

The SA scheduler uses the MCT and MET schedulers in a cyclic fashion depending on the load distribution across the machines. In this way, the SA tries to make benefit of the desirable properties of both MCT and MET. The MET heuristic can potentially create load imbalance across machines by assigning many more jobs to some machines than to others, whereas the MCT heuristic tries to balance the load by assigning jobs for earliest completion time. If the jobs are arriving in a random mix, it is possible to use the MET at the expense of load balance until it reaches a given threshold, and then use the MCT to smooth the load across the machines.

In Kim *et al.* (2003), in addition to MaxMin and MaxMax algorithms, the Percent Best scheduler was considered. The Percent best scheduler, which is a variation of the aforementioned k-Percent Best scheduler (KBP) (Maheswaran *et al.*, 1999), tries to map jobs onto the machine with the minimum execution time while considering the completion times on the machines. The idea behind this scheduler is to pick the top m machines with the best execution time for a job, so that the job can be mapped onto one of its best execution time machines. However, limiting the number of machines to which a job can be mapped, may cause the system to become

unbalanced. Therefore, the completion times are also considered in selecting the machine to map the job. The scheduler clusters the jobs based on their priority. Then, starting from the high priority group, for every job in this group, it finds the top m(=3) machines that give the best execution time for that job. Then, For each job, it finds the minimum completion time machine from the intersection of the m-machine list and the machines that are idle. For jobs with no tie, the mapping is performed immediately. For those jobs that are in a tie with some other jobs, that job that has earliest primary deadline is mapped first. The process is continued until all jobs in the high priority group are mapped. Then, the same procedure is applied to the jobs of other lower priority groups. They considered an increase in m when the priority of group decreases. In addition to the Percent Best scheduler, they introduced the Queuing Table, the Relative Cost, the Slack Suffrage, the Switching Algorithm, and the Tight Upper Bound (TUB) schedulers. The Queuing Table scheduler, which considers urgency in its mapping process, uses the Relative Speed of Execution (RSE), which is the ratio of the average execution time of a job across all machines to the overall average job execution time for all tasks across all machines, and a threshold to divide jobs into two categories of fast and slow. Using this categorization, and also estimating the nearness of the jobs deadline, the scheduler first maps those jobs that are in higher "urgency". The Slack Suffrage scheduler, which is a variation of the Suffrage scheduler Maheswaran *et al.* (1999), uses a positive measure of percentage slack of all jobs on all machines with various deadline percentages, and then maps those jobs with tighter deadline (higher deadline factor that was estimated for that job when enforcing positivity of the percentage slack measure). Please note that the Relative Cost scheduler does not have any direct relation with profit, and in fact the cost was defined based on the completion time. Cases of high and low heterogeneity and also tight and loose deadlines were also considered. It was observed that the Max-Max works the best in the high heterogeneity and loose deadlines cases, while the Slack Suffrage heuristic was the best in the low heterogeneity and loose deadlines cases. In those cases with tight deadlines, all schedulers showed low performance. Relatively, in the highly heterogeneous and tight deadlines cases, Max-Max and Slack Suffrage were better, while Queueing Table performed better in the low heterogeneity and tight deadlines cases.

1.1.2 Energy-Aware Scheduler

Global move toward ICT enabling effect, which pushes ICT to replace or dematerialize other sectors within upcoming decades, targets reducing human footprint and their impact on the environment (Liu *et al.*, 2011). At the same time, concepts, such as smart city and other smart initiatives, try to use artificial intelligence in order to reduce the cost and time of services while improving the quality of experience (QoE) of the users (Wright, 2012). All these moves depend highly on Compute as a Service and in particular High Performance Computing as a Service (HPCaaS) to handle spontaneous ICT requirements of service providers without forcing them to invest in capital. In this way, many service providers could spin off without the fear and limitations associated with capital expenditures of HPC computing facilities. Distributed data centers and in particular clouds could be a good approach to deliver HPCaaS at minimal cost and minimal environmental impact. However, ability to deliver HPC services on demand at an acceptable quality of service could be challenging. In addition, optimization of profit, expenditures, resource consumption, and environmental impact of such a solutions should be analyzed and verified.

Various work have been done to shed light on energy awareness in distributed data centers. For example, in Garg *et al.* (2011), a two-level broker to schedule jobs in a network of distributed data centers was proposed. They focused only on the HPC workload, and ignored constant-load web workloads. In their scenarios, they considered a set of data centers at different locations, and for each location they considered the average electricity grid mix carbon footprint and also the average electricity price. In addition, they assumed that dynamic voltage and frequency scaling (DVFS)-enabled processors used for the servers while having different minimum and maximum CPU frequencies at each data center. In terms of policy regarding the QoS, the jobs that could not be finished within the required deadline are dropped. Several greedy algorithms were included to reduce carbon footprint and increase the profit while meeting the required QoS. They concluded that carbon footprint can be reduced with negligible fall in the profit. Although the work is interesting, it suffers from various drawbacks. First of all, the electricity grid mix footprint and price are approximated with their average values, while in reality, these

footprint and price are highly variable and change even in an hourly scale. The assumption of average footprint and price prevented their algorithm to observe the electricity peak consumption phenomenon of the electricity grid. Peak hour management is a critical aspect to be considered in the design and management of any high-consumption facility. Moreover, in their DVFS-related optimization of the processors frequency, they obtained a constant optimal frequency for each type of CPU core which minimize the energy consumption of each individual job. However, this unpenalized DVFS-based approach could result in low performance in terms of HPC requirements and QoS. In other words, the optimal frequency is defined independent from the pricing and quality objectives, and it could not adapt to extreme cases when the footprint should be compromised in favor of quality of service or profit in order to ensure sustainability of the system operation.

Heuristic optimization has been also considered in many work for managing and scheduling HPC jobs and applications. For example, Kessaci *et al.* (2011) used a multiobjective approach using a GA algorithm to the scheduling of real HPC job traces on a distributed cloud. The solution was profit driven, and the cooling system was simply approximated using the Coefficient of Performance (COP) indicator. Similar to Garg *et al.* (2011), average values for electricity price and footprint, taken from EIA reports[1] were used. Also, the job deadlines were synthetically generated using the method proposed in (Venugopal *et al.*, 2008). They compared their results with those of maximum resource utilization heuristic. The main drawback of the GA optimizers, and any other heuristic optimizer used in the job scheduling, is that they cannot consider the complex and dynamic configurations of free slots in their formalism, and therefore usually end up to schedule only at the global level to the DCs. This condition highly simplifies the scheduling problem, and avoid maximum utilization of the detailed resources.

Consolidation of VMs on servers, or in general, any other type of application on servers, has been considered as a key action to reduce energy consumption and footprint of computing systems. With consolidation, the unused servers could be shutdown (to be more precise, the servers usually divided into three pools, the hot pool for those which are fully running, the warm pool for those servers which are reserved and are ready to join the hot pool in the case

[1] http://www.eia.doe.gov/cneaf/electricity/epm/table56a.html

of increase in the computing demand, and the cold pool that represents those servers that are completely shutdown). With shutting down those not-required-to-run-at-the-moment servers, all their associated idle energy consumption will be avoided, and also the lifespan of the servers would increase because they do not burn out in idle state. However, with recent progresses in manufacturing of more environment-friendly servers with very low idle consumption ratings, and also with price breakdown of the high performance processors, the idea of keeping all servers in the warm pool is getting more popular. This not only avoids many software originated faults that could be triggered in the shutdown and then cold start of the servers in the consolidation approach, it could also increase the lifespan of the server because of less physical/thermal stress being imposed on them. On top of this, ability to control the temporal performance of processors by adjusting the control voltage, which is usually referred to as DVFS, bring another dimension to the environment-friendly operation of the data centers. With DVFS, the servers consumption, which is mostly CPU consumption, could be adjusted and lowered by choosing lower operating frequencies, when the price or dirtiness of the electricity mix is high and the SLA and QoS do not impose very tight deadlines. In Feng *et al.* (2008), it was shown that by operating a supercomputer with low power processors and low power, not only the reliability of the system increases considerably with less down time (scheduled to replace dead processors), it also increases the relative performance to the space by three orders of magnitude. It was also shown that with applying a constraint on the maximum achievable performance of processors, performed by lowering the maximum performance by an epsilon (5% in that work), not only the energy consumption is reduced by a higher factor (20%), the processors were guaranteed that would not reach high temperatures, and therefore there were no processor failure because of high temperature breakdown.

DVFS approach has been considered in many other work as an alternative to shutdown and consolidation approach toward energy efficiency and reducing energy consumption. DVFS-based consumption reducing approaches rely on the nonlinear (usually cubic) relation between the control variable (frequency) and power consumption. A similar behavior based on nonlinear relations can be seen in many other components of a typical data center. For example, the fans used in the Computer Room Air Conditioner (CRAC) and Cooling Tower (CT) of the cool-

ing system also show a nonlinear polynomial relation between their control variable (relative performance compared to maximal achievable performance) and their energy consumption. In all these components, in analogy with the CPU processors, the energy consumption polynomially increases when the components operating state approaches its maximum nominal operating capacity. Although this can be compromised in some high-demanding applications where maximal operation capacity is always required, working in an intermediate state with relatively less consumption, cost, and footprint is a more practical choice for most of the use cases. In addition, even use cases that require higher capacities could be handled by increasing the number of operating components while operating them at a less-intensive state. It has been observed that this approach not only would give better performance in terms of consumption of the operation phase (ignoring the two other phases of life cycle: manufacturing and end of life), it is even more economical compared to high-intense solutions considering their extra capital expenditures associated with high rate of component replacements in intense solutions Feng *et al.* (2008). In other words, operating components at their maximum nominal rating requires frequent replacement of components that would result in an overhead capital cost distributed over the operating phase. Also, the inhomogeneity of the distribution of the surviving components along the time and also availability requirements of the solution would be much lower when components are operated at an intermediate rating because of less number of failures and in turn less number of replacement events. In addition, the requirement to operate at the near maximum operating rate forces only those components that have passed extreme "burn-in" tests to be used to reduce the number of replacements that in turn implicitly implies all those components that have failed and discarded during the burn-in tests should be considered in the calculations of the overall cost and footprint of the solution. That means that a solution designed at an intermediate-intense rating of components operation would be more available (reliable), less energy consuming, and therefore with less footprint.

Zhang *et al.* (2010) used DVFS to optimize the frequency of the running jobs to minimize the energy consumption of a system of heterogeneous cloud servers. However, they did not consider any other parameter which may play a role in the system energy consumption, performance, and profit. In another work, Rizvandi *et al.* (2010) argued that a time slack will be

produced after an optimal frequency is rounded up to the next possible CPU frequency of the DVFS scheme. Therefore, they breakdown the free slot in two pieces where the job will be run with two frequencies (maximum and minimum) instead of optimum frequency. However, they used a theorem[2] in their approach which is not in consistency with the Garg *et al.* (2011) claim of having an optimum frequency for a given job with minimum energy consumption.

Although manufactures and brands of CPUs and processors are usually ignored in the energy consumption studies, and instead a generic models is used, there are some studies that show proper choice of the CPU could lead to considerable savings. For example, in Nesmachnow *et al.* (2013), consumption, frequencies, operations, and also operations/watt of various CPUs are provided. In their model, they used four parameters to characterize a CPU: i) the computing power of a machine, i.e., the number of operations that its processor is able to compute; ii) the number of processing cores that the processor integrates (cores); iii) the energy consumption when the processor is in idle state (EIDLE); and iv) the energy consumption when the processor is fully loaded (EMAX). However, they did not consider the DVFS capability of the processors. For scheduling, they used various algorithms, such as Shortest Job Fastest Resource (SJFR) Abraham *et al.* (2000), Longest Job Fastest Resource (LJFR) Abraham *et al.* (2000), Opportunistic Load Balancing (OLB) Braun *et al.* (2001), Minimum Completion Time (MCT) Braun *et al.* (2001), Minimum Execution Time (MET), MinMin, MaxMin, and Suffrage that all focus on the makespan matrix and execution time of the jobs. In addition, they considered other algorithms, such as MINMIN, that perform similar logic to MinMin but with the energy consumption as the goal. Also, hybrid schedulers, such as MINMin, MINSuff, and MinMIN, were considered. In MINMin, the jobs and machines are first paired in such a way that the minimum completion time (MCT) is minimized. Then, in the second phase, those pairs are selected that minimize the energy consumption.

In Lawson and Smirni (2005), a power-aware scheduler was introduced. They observed a high degree of variability in the job arrival intensity across time (in week intervals) in various real workload traces. They also observed a high degree of variability in the jobs demand. These variabilities would result in variation in the utilization of the resources, especially existence of

[2]If f_a and $f_b(> f_a)$ execute a task in t_a and t_b, respectively. Then, $E(t_a) < E(t_b)$.

periods of time with very low utilization that would question the sustainability of the computing systems in terms of profit and costs. In Lawson and Smirni (2005), they used an indicator called the jobs bounded slowdown, defined as $1 + d/(\max(10, v))$, where d is the queuing delay time and v is the actual service time of a job. The 10 seconds in the denominator is for the sake of stability with respect to short jobs. They introduce a two-level policy that reduce the number of active processors to a lower value when the number of currently running processors goes lower than a switching threshold value, and bring back all processors to active state if the processing power required to handle incoming jobs goes higher than the same threshold value. They did not discuss the possible hysteresis side effect of having the same threshold for both actions. They chose their switching threshold value between 0.60 to 0.85 of the total active processors. The performance of the system was evaluated using the aggregated slowdown indicator. They observed that with their two-level policy, the aggregated slowdown was reduced by almost a factor of two, while 80% utilization of the computing systems and 10% saving was achieved. Therefore, this policy could provide 10% saving at 80% utilization if the user allows a twofold increase slowdown.

1.1.3 Profit-Aware Scheduler

In the real world of services and transactions, a computing service provider could survive and guarantee its sustainable operation if it has proper goals and strategies toward monetization and profitability of their solution (Sankaranarayanan *et al.*, 2011; Rao *et al.*, 2010), and also toward sustainable relations and good reputation with respect to their users (who are in turn service providers to the end users). Therefore, Profitability index (PI), also known as profit investment ratio (PIR) and value investment ratio (VIR), Quality of Service (QoS), and Quality of Experience (QoE) and other similar measures could play a critical role in management of such a solution which usually depends on its scheduler.

In (Toporkov *et al.*, 2011), several free slot selection algorithms, such as Algorithm based on Maximal job Price (AMP) and Algorithm based on Local Price of slots (ALP), were considered in connection with a profit goal. However, the proposed scheduler, which runs on a synthesized

simulator, does not consider the energy consumption or the carbon footprint associated with the operation of the NDC.

In Qureshi *et al.* (2009), a cost-aware request routing policy for Internet scale computing systems was introduced. The policy, which considers the variation of electricity price over time and location, preferentially maps the requests to those data centers that are cheaper. Because of Internet-scale nature of workloads considered in that work, the bandwidth price was also considered but in an abstract and indirect way by imposing a limit on the routing volume to keep it less than 95 percentile bandwidth of any location. To model the cooling system, they simply used a constant PUE value. Also, the idle power consumption was assumed to be a percentage (65%) of maximum power consumption. They showed that saving in electricity price is achievable if the electricity contracts are based on the actual power consumed not the provisioned power ratings.

1.1.4 Other type of Schedulers

Workflow scheduling is another type of scheduling that handles highly complex jobs (workflows). In Wu *et al.* (2013), scheduling of the workflows was considered using Directed Acyclic Graphs (DAG) modeling and also QoS constraints. In a DAG, each node represents a workflow task and directed links indicate task dependencies. To facilitate cloud workflow scheduling, each task node in a DAG is also associated with its QoS constraints. In that work, the execution time and execution cost were considered as the QoS constraints. The analyses were performed in a simulated environment, called Swinburne Decentralized Workflow for Cloud (SwinDeW-C), with 10 servers and 10 PCs. Several heuristic algorithms, such as GA, ACO, and PSO, were considered in different configurations. In the first configuration, only makespan was considered, and it was observed that when the number of tasks in the workflow is more than 200, ACO yields a better performance. This implies that the ACO is more effective in solving large size discrete multiple constraints optimization problem. One of the possible reasons for that could be because ACO constructs the valid solutions task by task while PSO and GA search for valid solutions randomly in the searching space. In a second configuration, both makespan and cost were considered, and it was noticed that each individual in ACO has its

social role for either makespan optimization or cost optimization. In GA and PSO, individuals are evolving to achieve higher fitness value but without any difference in social roles. It is well-known that time overhead of metaheuristics algorithm is their main concern. They observed that ACO CPU-time increases steadily with the growth of workflow size because it constructs and optimizes the solutions task by task. They also observed that most time consuming step of PSO is its update step. They also faced a premature problem with the GA algorithm for large workflows that prevents this algorithm from achieving the best score.

In Le *et al.* (2010), another scheduling of Internet-based workload was proposed. For the Internet scale applications that handle the Internet users' requests, the distributors usually send every request to two or more data centers at the same time as a mirroring technique in order to guarantee availability and performance. The request distribution policy was designed to prevent data center overloads, and also to monitor their response times, and adjust the request distribution to correct any performance or availability problems. In the modeling, they used a two-term cost representing the cost associated with the requests (SLA) and cost associated with green and brown energies. For the brown energy, they considered a cap limit below that they do not consider the market cost of offsetting the base energy of a data center. They used a Simulated Annealing (SA)- and a Cost-Aware (CA)-heuristic schedulers. Also, an Autoregressive integrated moving average (ARIMA) model was used to predict the upcoming load. They observed that diversity in parameters such as workload and electricity price, helps the system manager to achieve more savings.

A realistic approach to HPC scheduling should consider both profit and QoS indicators in its methodology in order to ensure sustainable operation of the NDC. For example, Goiri *et al.* (2012) proposed a profit-driven virtualized data center in which SLA and its associated penalties were implemented. In addition, they considered VM performance degradation. The proposed scheduler was a global scheduler for the VMs that considers VM consolidation as its approach to energy efficiency. Various scheduling policies, such as backfilling, random, and cost-driven were considered. They simulated the solution on a simulator built based on OMNet++ using real measurements. The simulator was validated in (Berral *et al.*, 2010; F. Julià,

2010). For the job trace, they considered one week excerpt of Grid5000 trace. In that one week, jobs had various duration: short (less than 600 seconds), medium, and long (more than 20,000 seconds). The jobs were distributed among time in such a way that a mean of 35 and a maximum of 204 tasks are running concurrently. Also, they considered the effect of different types of servers, such as Xeon, Atom, and their heterogeneous combination, and showed that the heterogeneous approach, which reduces the power consumption using the Atom power-efficiency and gets reasonable SLA fulfillment thanks to the Xeon nodes, results in higher savings.

In general, the GA-based approaches to scheduling are powerful in exploring the full space in their scope. However, their scope is very limited because of their intrinsic restrictions on definition of chromosomes (Goiri *et al.*, 2012; Kołodziej *et al.*, 2012). While in an actual scheduling, the free time slots could dynamically move across the servers and datacenters, the genes in the GA chromosome, and other similar approaches, require a persistent definition, and therefore, could not be modified during the scheduling operation. This has forced many of the GA-based scheduling work to adapt a global level and only consider scheduling down to the level of datacenters, and leave the actual scheduling of the jobs at the server level to local schedulers. This limitation reduces the opportunistic ability of these schedulers because they could not observe local opportunities and work only at the global picture of the NDC.

Another example of a GA-based approach to scheduling is Kołodziej *et al.* (2012). In this work, a detailed DVFS is considered for the processors' power consumption. In addition, various scheduling scenarios against various schedulers, such as GA and island GA, were considered. The scheduling was based on the makespan matrix. They used the HyperSim-G software for their simulations.

In Table 1.1, features of the state-of-the-art schedulers described in this section are summarized. The acronyms M.S., S.S., H.S., E.P., E.M., E.V., E., C.E., C.T., P., Pr., Q., C., D., Di., and Co. stand for Meta Scheduler, Super Scheduler, Heuristic Scheduler, Electricity Price, Electricity Mix, Electricity Variations, Energy, Carbon Emissions, Carbon Tax, Profit, Prediction, QoS, Consolidation, DVFS, Distributed system, and Cooling system, respectively.

Table 1.1 The comparison table among state-of-the-art approaches.

Paper	Scheduler			Electricity			E.	Carbon		P.	Pr.	Q.	CPU Power		Di.	Co.
	M.S.	S.S.	H.S.	E.P.	E.M.	E.V.		C.E.	C.T.				C.	D.		
Wu *et al.* (2013)	✓		✓							✓						
Braun *et al.* (2001)	✓		✓													
Kessaci *et al.* (2011)	✓		✓	✓			✓	✓		✓				✓	✓	✓
Garg *et al.* (2011)	✓			✓			✓	✓		✓				✓	✓	✓
Goiri *et al.* (2012)	✓						✓			✓		✓	✓			
Maheswaran *et al.* (1999)		✓														
Kim *et al.* (2003)		✓														
Freund *et al.* (1998)		✓														
Nesmachnow *et al.* (2013)		✓					✓									✓
Toporkov *et al.* (2011)		✓								✓		✓				
Kołodziej *et al.* (2012)		✓	✓				✓							✓		
Le *et al.* (2010)			✓	✓				✓			✓					
Guzek *et al.* (2012)			✓											✓		
Lawson and Smirni (2005)												✓				
Qureshi *et al.* (2009)				✓		✓				✓						
Feng *et al.* (2008)							✓				✓					
Our research		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

1.2 Server Consolidation and Load Balancing in Cloud Computing

Energy efficiency and carbon footprint reduction are two main concerns in the design and development of computing systems (Beloglazov *et al.*, 2010). Energy efficiency is, in fact, imperative, with the increasing cost of energy and the need to reduce GhG emissions (Berl *et al.*, 2010). There are several ways to achieve energy efficiency in data centers, such as dynamic CPU speed management, energy aware job scheduling, and server consolidation (Zhang *et al.*, 2008; Srikantaiah *et al.*, 2008).

Server consolidation has been studied using various approaches in data centers. Bianca Pop *et al.* introduced swarm-based consolidation for data centers in (Pop *et al.*, 2012). Their approach was inspired by the V formation that birds adopt when flying to achieve maximum energy efficiency. In another method, a "gossip"-based methodology is used, which is rooted in a communication and negotiation technique among neighboring servers (Marzolla *et al.*, 2011).

With technological advances in virtualization technology, it is possible to run different servers with different platforms on a single physical machine in a completely isolated environment. This will reduce the number of hardware needed in a data center which will directly lead to

energy efficiency, carbon footprint reduction, and operational cost reduction. Server consolidation can be seen as a bin packing problem, where servers are the bins (S_i, $i = 1, 2, ..., m$) with different capacities in CPU (S^{cpu}), memory (S^{mem}), network (S^{net}), and storage (S^{hdd}). VMs represent the items (V_i, $j = 1, 2, ..., n$) which need to be fit in the bins with their required capacities in CPU (V^{cpu}), memory (V^{mem}), network (V^{net}), and storage (V^{hdd}). When each VM is assigned to a server, the objective is to minimize the number of servers (bins), where following conditions are satisfied:

$$
\begin{aligned}
&minimize\ |\mathbf{S}|\,, \\
&\sum_{V_k \in S_i} V_k^{cpu} \leq S_i^{cpu},\ S_i \in \mathbf{S} \\
&\sum_{V_k \in S_i} V_k^{mem} \leq S_i^{mem},\ S_i \in \mathbf{S} \\
&\sum_{V_k \in S_i} V_k^{net} \leq S_i^{net},\ S_i \in \mathbf{S} \\
&\sum_{V_k \in S_i} V_k^{hdd} \leq S_i^{hdd},\ S_i \in \mathbf{S} \\
&\bigcup_i S_i = \{V_1, V_2, ..., V_n\},\ S_i \in \mathbf{S}
\end{aligned}
\tag{1.1}
$$

where $\mathbf{S}$ is the set of all active servers, and $|\mathbf{S}|$ is the cardinality of that set (which is also equivalent to the number of active servers).

In general, a simple bin packing problem can be described as follows:

$$
\begin{aligned}
&minimize\ |\mathbf{B}|\,, \\
&\sum_{I_k \in B_j} i_k \leq b_j,\ B_j \in \mathbf{B} \\
&\bigcup_j B_j = \{I_1, I_2, ..., I_n\},\ B_j \in \mathbf{B}
\end{aligned}
\tag{1.2}
$$

where $\mathbf{B}$ is the set of all active bins. In addition, I and i represent an item and its corresponding sizes, B and b represent a bin and its corresponding sizes, and n represent the number of items respectively.

The server consolidation can be performed with First Fit (FF) algorithm or any of its many variations built to solve energy efficiency problems (Beloglazov *et al.*, 2010), which are basic bin packing solutions such as a potential hardware cost savings model in (Speitkamp and Bichler, 2010), the vector packing model in (Gupta *et al.*, 2008), and a model for the Mixed Integer Program (MIP) in (Petrucci *et al.*, 2009). Most of these models define server consolidation problem as a form of the bin packing problem, and attempt to maximize the use of servers with intuitive algorithms and methods, such as the Best Fit and First Fit Decreasing (FFD) algorithms. There are two main categories for server consolidation: static and dynamic. In static server consolidation, a VM location is decided prior to its creation, and after VM creation, the VM will be hosted on the same hardware until its removal. On the other hand, in dynamic server consolidation, VM management controllers take advantage of seamless VM migration technology and move the VMs from one hardware to another without service interruption. Heuristic algorithms introduced in the following section and our proposed algorithm are all designed for dynamic server consolidation problems.

From another perspective, GGA is also used for server consolidation problem. GGA, similar to First Fit Decreasing algorithm, Best Fit algorithm, any many more of their variations try to solve the server consolidation problem in a data center as a bin packing problem. However, we are specifically interested in GGA algorithm in this thesis because of its grouping feature. Since in this research we are mainly working on network of data centers, we describe the DCs as high-level groups, and servers as low-level groups. In fact, from our perspective, the servers are observed as subgroups of DC groups. This is achieved, in the Section 5.1, an extension of GGA is introduced to deal with the NDC consolidation problem.

Here, first, we present some of the work in the area of server consolidation with GGA, and next, we present the mechanism of grouping in GGA algorithm which later will be used to introduce the new variation of GGA algorithm for higher levels of grouping.

1.2.1 Grouping Genetic Algorithm in Server Consolidation

Xu et al. used Grouping Genetic Algorithm (GGA) in (Xu and Fortes, 2010) in order to achieve multi-objective goals in placement of virtual machines in virtualized data center environments. They claimed that original GGA crossover operator is not efficient and they modified it to achieve better results. They proposed a ranking-crossover instead, and claimed that new crossover is able to inherit good features from parents more efficiently. They evaluated all the individuals based on three evaluation functions which they used to represent their three optimization objectives. These three objectives were resource usage efficiency, power consumption efficiency, and thermal efficiency. They represented some evaluation functions for each of these objectives. The evaluation results were some numerical values in the interval of [0,1]. Instead of random selection of crossover points, the selected groups for insertion to the first chromosome are most likely selected from groups with higher rank in ranking evaluation of three objectives. They claimed that this way, the high quality groups will most probably remain intact, and therefore the optimizer will reach to a better solutions faster. They also combined GGA with fuzzy concepts in order to achieve the best solution for their several objectives problem.

Shubham Agrawal et al. used the GGA algorithm for a server consolidation problem in Agrawal *et al.* (2009). They modeled the server consolidation problem as a vector packing problem with conflicts. In their mathematical model, they tried to differentiate between efficiency of bin packing and number of bins which are packed. Their model was designed to prefer the bin-packing efficiency over bin number optimization. They used the original version of the GGA in order to solve the optimization problem.

In another work (Wilcox *et al.*, 2011), David Wilcox et al. introduced another type of GGA algorithm known as Reordering Grouping Genetic Algorithm (RGGA). They describe the multi-capacity bin-packing problem in data center server consolidation as bins (servers) with multiple capacities (CPU, memory, network, storage, and etc.) and VMs with multiple weights. In their proposed grouping genetic algorithm, each individual has several representations, and they claim these multiple representation will lead to better solution in more efficient time frame.

Parent chromosomes are chosen with a higher probability from those individuals with higher fit. In their approach, they combined all the bins from both parent chromosomes and sort them by fitness. The fuller a bin is, it is on top of the list, and therefore less full bins are at the bottom of the list. From the top of the list, some bins will be selected and the rest of the bins will be discarded. If there is a bin which contain an individual belongs to already selected bins, that bin will be discarded as well. For the individuals which are discarded, they will be ordered by their fitness and first fit descending algorithm will be used in order to reinsert them to the offspring chromosome.

Because the algorithm always prefers tightly packed bins over other bins, they added a Gaussian noise to the fitness function of the individuals in order to escape the local minimums. Respectively, in their mutation operator, the mutation take place more on less fit bins than good bins. This will assure that the structure of good groups does not intact often. They used three mutation operator. First one is the normal GGA mutation in which some bins will be randomly removed, and their associated individuals will be reinserted into the other bins. In the second method, two items in the order list will be swapped, and finally in the third one, one item will be randomly relocated in the order list.

1.2.2 Grouping Mechanism in Grouping Genetic Algorithm

In Falkenauer and Delchambre (1992), Falkenauer and Delchambre proposed a new version of genetic algorithm known as grouping genetic algorithm. They argue that normal genetic crossover and mutation operators are not able to preserve the group features of the parent chromosomes. In the straightforward encoding scheme, each item (for example, a VM) is represented by a gene in the chromosome, and its label is its group (for example, a server) which that item belongs to. For example, the chromosome ADEBFFBC encode a solution for 8 VMs where the first VM is on server A, the second VM is on server D, and so on. Basically, when there are two parents with good groups defined in their chromosomes, there is no way for normal genetic crossover operator to create an offspring in which those good groups are preserved. A part of a child chromosome comes from one parent, and the rest comes from the other parent. Therefore, well-defined groups in both parents will break in parts, and the

probability of having an offspring with stronger groups is very low. Therefore, they proposed a new crossover and mutation operators in their new algorithm, which perform on groups instead of individual genes.

In their crossover operator, the groups presented in the chromosomes are lined up (keeping one gene per group), and the crossover will happen on these two group representations of the parents. For example, for the chromosome ADEBFFBC, the group lineup will be ADEBFC. It is worth noting that, in the group representation, the chromosomes could be of variable length. Two crossover points will be randomly selected in each parent group-lineup. And, the groups in middle part of the second parent group-lineup will be inserted in first parent group-lineup at the first crossover point. For example, consider a group-lineup of the parents as follows:

P1 : ADE||BFC (ADEBFFBC)
P2 : bd|ca| (bbdcabba)

where the groups with same alphabetic character but with different cases (upper and lower cases) are same but represent that group in first and second parent, and crossover points are marked as |. Also, the straightforward encoding of the chromosome is provided in parentheses.

After insertion, the offspring group lineup of the offspring will look like (ADEcaBFC). Because the groups “c” and “a” are inserted from the second parent, their matched groups in first parent “C” and “A”, are no longer valid and these two groups and all their assignments to individual genes will be removed from the offspring; remaining the offspring group lineup as (DEcaBF). For our example, the straightforward encoding of the offspring will be: (?DEcaFBa). "?" symbol shows that the first individual gene has no group assigned to it any more because group A is removed from the chromosome. In a same way, there are some individuals which are in groups “c” and “a” in second parent while they are in other groups in first parent. The group of these individuals will be replaced with inserting groups from second chromosome. The groups of replaced individuals need to be removed with all assignment to individual genes which are groups “B” and “F” in the first parent.

For our example, the straightforward encoding of the offspring will be: (?DEca??a). Now, there are some individuals which their group assignments are removed from chromosome in previous actions which needs to be reinserted in the offspring chromosome. First Fit Descending algorithm (Garey and Johnson, 1979) is used in order to reinsert the removed individuals into the chromosome. The priority is with the groups which are almost full.

In Figure 1.1, another example is illustrated with larger variety of groups. Each individual gene is represented by a number. Groups are represented by circles and they are labeled by alphabetic characters. Circles with same color are in a higher level group.

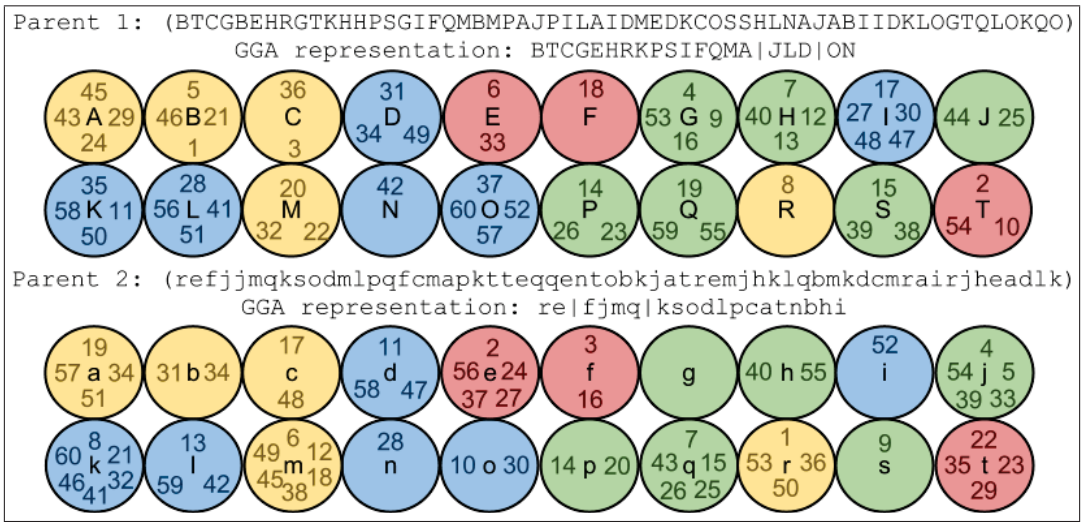


Figure 1.1 GGA representation for parent chromosomes.

The crossover operation in progress is illustrated in Figure 1.2. The arrows indicate the new position of the genes in the offspring (child) chromosome. Red arrows represent the genes coming from replaced groups. Black arrows represent the genes from removed groups, and blue arrows represent the genes from other groups.

Some genes are removed from the chromosome which are indicated by a cross sign. The black cross sign indicate the removed genes in a removed group. Red cross sign indicate the removed genes in a replaced group, and blue cross sign indicate the removed genes in other group with at

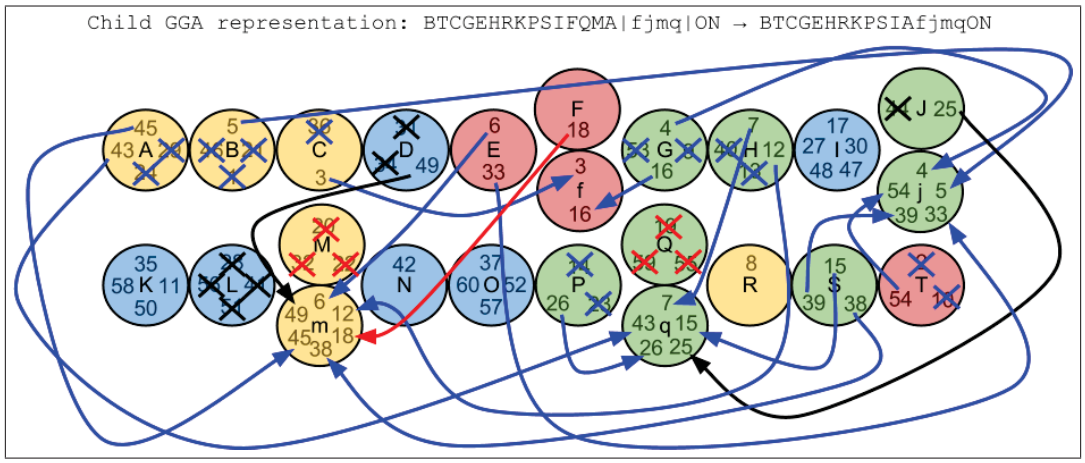


Figure 1.2 GGA crossover in progress.

least one gene in inserted groups from second parent chromosome. The final child chromosome is illustrated in Figure 1.3.

In mutation operator of grouping genetic algorithm, the lineup of groups will be created in a similar way of the crossover operation. Then, some groups will be chosen by random and those groups with their containing individuals will be removed from the chromosome. Then, there are some individuals, which have been removed in previous action, and are needed to be reinserted into the chromosome. A similar action as that of the crossover operator will be taken here in order to reinsert the removed individuals into the chromosome.

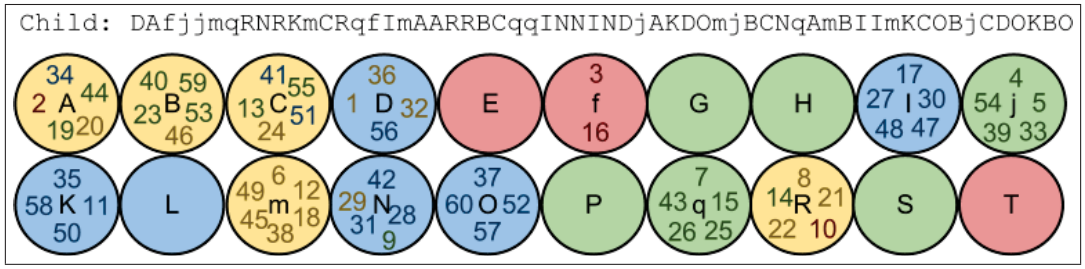


Figure 1.3 GGA crossover final result.

1.3 Server Energy Metering

Although higher server utilization is one of the possible motivations for server consolidation, data centers have other important objectives, such as energy efficiency and carbon footprint reduction, which should be considered. This requires accurate models for provisioning the energy consumption or carbon footprint of the servers.

The basic model for the energy metering of a physical server was introduced in Kansal *et al.* (2010):

$$E_{\text{sys}} = \alpha_{\text{cpu}}\mu_{\text{cpu}} + \alpha_{\text{mem}}\mu_{\text{mem}} + \alpha_{\text{io}}\mu_{\text{disk}} + \gamma \tag{1.3}$$

in which the total energy consumed by a physical machine E_{sys} is calculated based on the rate of use of various elements of that physical machine, namely: CPU (μ_{cpu}), memory (μ_{mem}), and disk usage (μ_{disk}). α_{cpu}, α_{mem}, α_{io}, and γ were the equation constants. In this work, the authors claimed that the model has a small margin for error and low processing overhead. However, they assumed that it is suitable for a virtualization environment without validation (Bertran *et al.*, 2010a).

Bertran *et al.* then introduced a model to measure the power consumption of the CPU and memory of a server based on performance monitoring counters (Bertran *et al.*, 2010b):

$$P_{total} = \sum_{j=1}^{cores} \left\{ \left(\sum_{i=1}^{comps} AR_{ij} \times P_i \right) + P_{static} \right\} \tag{1.4}$$

where P_{total} is the total power consumption of the CPU and the system's memory components in a multicore server, AR_{ij} represents the Performance Monitoring Counter (PMC)-based formula accounting for the activity ratio, and P_i are the formula constants. These constants can be calculated through linear regression techniques for each type of server from experimental data. In that work, the same model was used and validated for virtualized environments (Bertran *et al.*, 2010a). Despite the accuracy of the model, in order to arrive at the total energy con-

sumption of a virtual machine, a server, or the whole cloud, components other than the CPU and the memory in a hardware machine, such as network, storage, and utilities, need to be taken into consideration.

In Liu *et al.* (2009), the energy consumption of a cloud is divided into a number of terms:

$$E_{total} = E_{migration} + E_{servers} + E_{utilization} \tag{1.5}$$

where E_{total} is the total energy consumption of the data center, $E_{migration}$ is the energy consumption corresponding to migration of the virtual machines (VMs), $E_{servers}$ represents the energy consumption of the servers, and $E_{utilization}$ represents the energy consumption of other utilities. Note that the energy consumption of VMs is normally included in the energy consumption of the physical machines. However, the migration cost reflects the extra energy consumption of physical machines, like servers and routers, required to migrate a VM from a source server to a destination server.

If the frequency of the CPU cores of a server is adjustable between a minimum and a maximum value, the power consumption of the CPU varies significantly with variations of CPU frequency. The power consumption of CPU can be presented as a function of frequency (Wang and Lu, 2008; Chen *et al.*, 2005):

$$P_{\text{cpu}} \quad = \quad \beta_{\text{cpu}} + \alpha_{\text{cpu}} f_{\text{cpu}}^3 \tag{1.6}$$

In summary, Equations (1.3) and (1.4) provide two ways for calculating the power consumption of servers. These two models are compared in the experimental results section with our proposed model. Our proposed model for server power metering is used to calculate the power consumption of servers in the simulation platform. Equation (1.5) considers an extra term for power consumption of data centers that represents the power consumption of the migration

actions which are used in our load balancing experiments. Last, Equation (1.6) is used for calculating the energy consumption of fully-utilized servers in the HPC job scheduling scenarios.

1.4 Cooling System Power Modeling

The main part of supporting facilities in any data center is its cooling system. Traditionally, the cooling system was used to consume as much energy as the main IT equipments. However, with the general move toward efficiency and lowering costs and also environmental footprint of data centers, the cooling systems are also becoming smarter and more efficient (Wright, 2013). It worth noting that, even at hypothetically zero-footprint operation, i.e., being carbon neural, the issue of wasted heat generated by the cooling systems (and in general the whole data center) will be becoming a critical factor in the design of next generation of data centers. This is a result of the IT industry move toward a more environmentally responsible operation in alignment with the ICT enabling effect. In this section, we review the cooling systems and also their modeling.

Because of complexity of the energy consumption in data centers, which is heterogeneously distributed among various components at different levels and scales, a reliable operation of data center highly depends on identification, reduction, and handling of the heat generated across the whole spectrum of components involved: from chips, blades, racks, and computer rooms (CRs) to the cooling system itself. Although heat handling at the chip level is usually unnoticed, it has been observed that almost 30% of the cooling power consumption is related to this level of cooling to feed semiconductor fridges and also fans in order to extract the heat from the chips (Patel *et al.*, 2006). It worth noting that this portion of cooling system consumption is somehow hidden to the analyzers and does not counted in the actual analysis of the data centers, such as in the Power usage effectiveness (PUE) analysis. The PUE is defined as the ratio of the total power consumption of the data center facility to that of the IT equipments

(Haas *et al.*, 2009):

$$\begin{aligned} \text{PUE} &= \frac{\text{Total power consumtion}}{\text{IT power consumption}} \\ &= \frac{\text{Support facilities consumption} + \text{IT consumption}}{\text{IT consumption}} \end{aligned} \tag{1.7}$$

where Support facilities power consumption is the total power consumption of the support facility equipments, such as cooling system, power distribution system, and lighting system. A recommended practice for measuring the PUE is the weekly-averaged PUE using data points gathered continuously with a Level 2 (at the PDU level) meter placement, and with the period of measurement and assessment of a year (Haas *et al.*, 2009; Tipley, 2012). The annual condition helps to avoid seasonal variations in the assessment.

It is worth noting that the cooling systems are usually designed with oversized capacities. The designers are required to consider oversized cooling systems in order to guarantee redundancy and also allow future additional installation of new IT equipment. In general, oversizing helps to avoid frequent downtime with preventing: 1) condensation, 2) temperature-related failure, and 3) cold installation of additional IT equipment.

In this study, we will use with an air-cooled, raised-floor, bricks-and-mortar (is often used to refer to a company that possesses a building for operations) datacenter architecture with a chiller cooling plant. In this section, a review on the elements and also related models is presented. A cooling system can be divided into three major parts: 1) the computer room AC (CRAC), 2) the chiller facility, and 3) the cooling tower. Air/liquid/air systems has the advantage of ability to rapidly extract the heat from the facility; one kilogram of water can store about four times as much thermal energy as the same mass of air, while its volume is much smaller.

The energy consumption of the cooling system of a typical datacenter is a very complex process because many heterogeneous and completely different components play essential roles in a tight collaboration. From the chip scale to the cooling tower scale, energy consumption and also heat handling are very critical and at the same time they rely highly on the performance

at other scales. To be precise, we assume that there are 6 scales in a data center: 1) chip, 2) blade, 3) rack, 4) computer room (CR) (also, we may unofficially call it data center), 5) CR + chiller units (CH), 6) CR + CH + cooling tower (CT). The components used at each scale and also the physics that governs its related phenomena are completely different. As our study is more focused on the management of a distributed cloud comprised of a few data centers, we assume that the same best practice has been used for heat handling at the chip, blade and rack level in all data centers. Therefore, we will ignore the impact of heat handling at these scales in our models. However, we want to point out these scales play a critical role in the total cooling performance of a datacenter. For example, as mentioned before, fans and thermoelectric coolers at the chip level can consume as high as equivalent of 10% to 30% of the power of the chip itself (Patel *et al.*, 2006). Also, the design of the air flow inside and around a rack (for example, using snorkels) can impact the severity of hot spots and temperature distribution within racks, which in turn impacts the amount of cooling power or the inlet temperature required (Das *et al.*, 2010).

The high-level schematic of the datacenter in terms of power consumption and also heat handling is shown in Figure 1.4. The heat generated by the IT equipments, the lighting, and also the CRAC's fans themselves is removed from the CR by the CRAC units via an air loop. The warm air collected by CRAC units is passed through heat exchangers coupled with the chilled water loop of the chiller plant. The warm water is passed to the chiller units where, in a mechanical refrigeration cycle, the heat is transfered to another water loop connected to the cooling towers. In the chillers part, several pumps condense the water in the cooling tower loop, and some other secondary pumps move the chilled water in the chilled water loop connected to the CRAC units. In the cooling towers, some fans create a vaporization cycle of outdoor air to extract heat from the warm water. In the following sections, energy consumption and heat handling of each component of the cooling system are separately presented.

In the following sections, we follow the same path as that of Figure 1.4 to analysis the components of a data center.

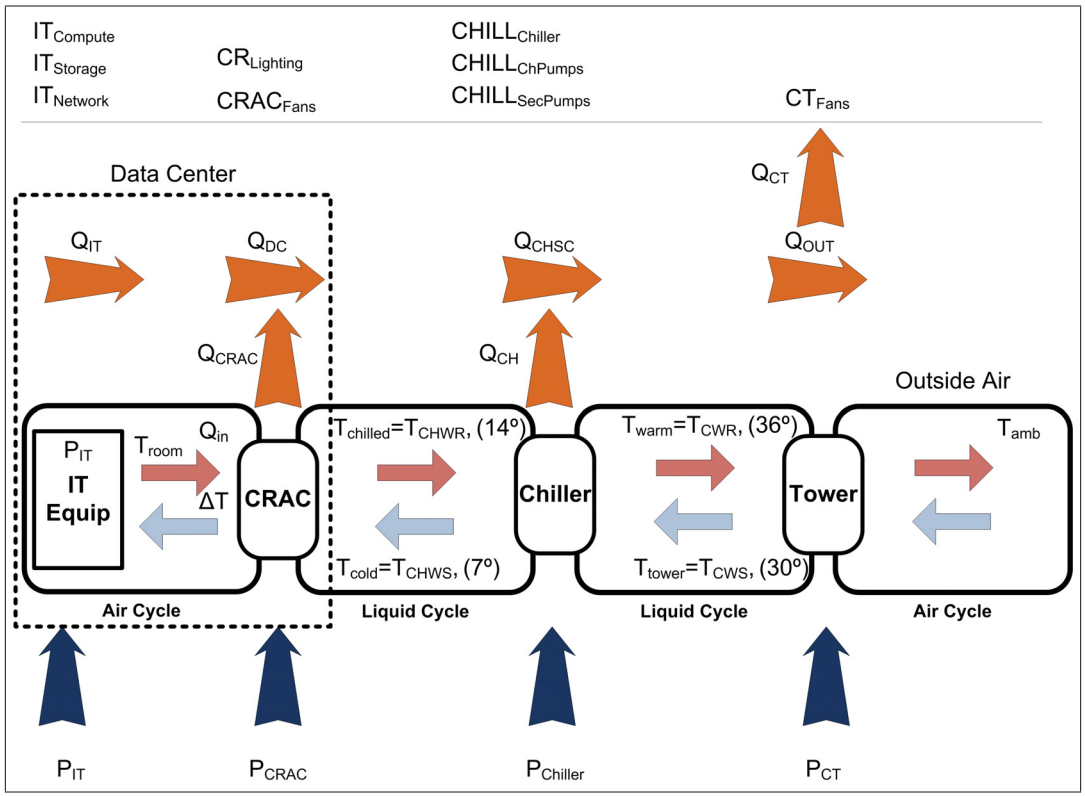


Figure 1.4 The chiller plant (cooling system) overview.

1.4.1 Computer room (CR)

In a computer room, there are 4 components that consume energy: 1) IT equipments. 2) lighting, 3) humidity controller, and 4) CRAC fans.

IT equipments, Lighting and Humidity Control

The power consumption of IT equipments are discussed in the previous section (refered here by P_{IT}), and although lighting and humidity control can considerably contribute to energy consumption and also heat generation, we assume that their power consumption and heat generation are zero:

$$P_{\text{Lighting;Humidity}} = 0,\ Q_{\text{Lighting;Humidity}} = 0 \tag{1.8}$$

CRAC units

The main component of a CRAC is its fan(s) that moves cold air toward racks and collects the warm air. We assume all fans are equipped with variable-frequency drives (VFD) that allow real-time control of fans speed. The power consumption of CRAC fans can be modeled as an almost cubic (power of 2.75) relation (Das *et al.*, 2010):

$$P_{\text{CRAC}} = P_{\text{CRAC,fan}} = P_{\text{CRAC,fan,max}}\theta_{\text{CRAC}}^{2.75} \tag{1.9}$$

where θ_{CRAC} is the utilization ratio of the CRAC (between 0 and 1). A typical value for $P_{\text{CRAC,fan,max}}$ is 6.32 kW. Usually, more than one CRAC is installed in a CR. However, we did not explicitly put the number of CRAC units n_{CRAC} in the above equation to emphasize on the possibility of running each CRAC at a different utilization ratio. However, from here on and for the purpose of simplicity, we assume that all CRAC units in a CR are working at the same utilization ratio.

The air flow generated by the fans is also affected by the value of θ_{CRAC}:

$$\phi_{\text{CRAC}} = \phi_{\text{CRAC,max}}\theta_{\text{CRAC}} \tag{1.10}$$

This equation will be used later in calculating the heat handling capacity of the CRAC units in the next section. A typical value for $\phi_{\text{CRAC,max}}$ is 5.85 m^3/s (Das *et al.*, 2010).[3]

1.4.2 Chillers

The total energy consumed by the chiller plant, P_{CH}, is equal to:

$$P_{\text{CH}} = P_{\text{chillers}} + P_{\text{ChPumps}} + P_{\text{SecPumps}} \tag{1.11}$$

[3] 1 m^3/s = 2119 cfm where cfm stands for cubic feet per minute.

where P_{ChPumps} and P_{SecPumps} are the power consumption of chiller's pumps and chiller's secondary pumps respectively, and

$$P_{\text{chillers}} = P_{\text{ChComp}} + P_{\text{ChCondFan}} \tag{1.12}$$

where P_{ChComp} is the power consumption of chiller's compressors and $P_{\text{ChCondFan}}$ is the power consumption of chiller's condensation fans. We will only use P_{chillers} in our modeling from here on.

If we define P_{chillers} as chiller power consumption at a specific θ_{Chiller} value (the utilization ratio of the Chiller), we have (Lee and Lee, 2007):

$$P_{\text{chillers}} = P_{\text{chiller,max}} n_{\text{Chiller}} \left(A_{\text{Chiller}} \theta_{\text{Chiller}} + B_{\text{Chiller}} \theta_{\text{Chiller}}^2 \right) \tag{1.13}$$

where A_{Chiller} and B_{Chiller} are the chiller's parameters. We use the following typical values in this study (Lee and Lee, 2007): $A_{\text{Chiller}} = 0.3799$ and $B_{\text{Chiller}} = 0.6194$. n_{Chiller} is the number of chillers planned in the chiller plant. Having more than one chiller not only allows to save more energy, it can reduce the failure risk because of higher redundancy. For example, for $\theta_{\text{Chiller}} = 75\% = 0.75$, we have $P_{\text{chiller}} = 0.6333 n_{\text{Chiller}} P_{\text{chiller,max}}$.

Chiller pumps

Usually, because of high level of design dependency, a pump-chiller ratio (PCR) is considered to relate the pump power consumption to the chiller power consumption in chiller units. Based on typical systems in the market, the PCR is around 0.3 (Energy Design Resources, 2010). In cases where there are more than one chiller (with one pump per chiller), the total power consumption of the pumps is:

$$\begin{aligned} P_{\text{ChPumps}} &= PCRP_{\text{chiller,max}} n_{\text{ChPumps}} \times \\ &\quad \left(A_{\text{ChPump}}(\theta_{\text{ChPump}}) + B_{\text{ChPump}}(\theta_{\text{ChPump}})^2 - C_{\text{ChPump}}(\theta_{\text{ChPump}})^3 \right) \end{aligned} \tag{1.14}$$

where $n_{\text{ChPumps}} = n_{\text{Chiller}}$ is the number of pumps. θ_{ChPump} is the utilization ratio of a chiller pump. Assuming the same number of pumps as the number of chillers:

$$\theta_{\text{ChPump}} \simeq \theta_{\text{Chiller}}. \tag{1.15}$$

$A_{\text{ChPump}} = 0.338249$, $B_{\text{ChPump}} = 0.972488$, and $C_{\text{ChPump}} = 0.291451$ for a typical system (Lee and Lee, 2007).

The mass flow of water in the loops is assumed to be a function of pump utilization:

$$(\mathrm{d}m/\mathrm{d}t)_{\text{CHW}} = (\mathrm{d}m/\mathrm{d}t)_{\text{CHW,max}}\theta_{\text{ChPump}} \tag{1.16}$$

$$(\mathrm{d}m/\mathrm{d}t)_{\text{CW}} = (\mathrm{d}m/\mathrm{d}t)_{\text{CW,max}}\theta_{\text{ChPump}} \tag{1.17}$$

where $(\mathrm{d}m/\mathrm{d}t)_{\text{CHW}}$ and $(\mathrm{d}m/\mathrm{d}t)_{\text{CW}}$ are the water mass flow in the chiller-water loop and cooling tower-water loop, respectively.[4] The $(\mathrm{d}m/\mathrm{d}t)_{\text{CHW,max}}$ will be calculated in the next section.

1.4.3 Cooling tower (CT)

A cooling-tower chiller ratio (CCR) is considered to relate the cooling tower power consumption to the chiller power consumption in a chiller plant. Based on typical systems in the market, we assume a CCR of 0.12 (Energy Design Resources, 2010). The CT fans follow a similar governing equation as that of the CRAC units with different parameter values (assuming variable-frequency drive (VFD) fans):

$$P_{\text{CT}} = \text{CCR}\ P_{\text{chiller,max}} n_{\text{Chiller}} \theta_{\text{CT}}^{2.75} = P_{\text{CT,max}} \theta_{\text{CT}}^{2.75} \tag{1.18}$$

where θ_{CT} is the utilization ratio of the CT. A typical value for $P_{\text{CT,max}}$ will be $0.12 \times 80 \times 5 = 48\text{kW}$.

[4] CHW and CW stand for chiller-water and cooling tower-water respectively.

The air flow of the CT fans is also affected by the θ_{CT}:

$$\phi_{\text{CT}} = \phi_{\text{CT,max}}\theta_{\text{CT}} \tag{1.19}$$

A typical value for $\phi_{\text{CT,max}}$ is calculated as:

$$\begin{aligned} \phi_{\text{CT,max}} &= \frac{\phi_{\text{CRAC,max}}}{P_{\text{CRAC,fan,max}}}\text{CCR}\ P_{\text{chiller,max}}n_{\text{Chiller}} & (1.20)\\ &= \frac{5.85}{6.32}0.12 \times 80 \times 5 = 44.42\ \text{m}^3\text{/s} & (1.21) \end{aligned}$$

1.4.4 Heat Handling Capacity in a Datacenter

CRAC units

The cooling power, or heat handling capacity, of the CRAC units can be expressed as follows:

$$P_{\text{cooling,CRAC}} = A_{\text{CRAC,cooling}}\phi_{\text{CRAC}}n_{\text{CRAC}}\left(T_{\text{CR}} - T_{\text{CHWS}}\right) \tag{1.22}$$

where T_{CR} is the ceiling temperature in the CR, and T_{CHWS} is the chiller water source temperature. n_{CRAC} is the number of the CRAC units. The specific heat capacity of air $A_{\text{CRAC,cooling}}$ is $1/(0.8634) = 1.1582$ in (kW/ (m^3/s C°)) and $1/3293$ in (kW / (cfm F°)). In this study, we assume $T_{\text{CR}} = 40°$ and $T_{\text{CHWS}} = 7°$. This comes to 227 kW cooling power for a typical CRAC unit when working at its maximum capacity. In this study, for a typical 1MW CR, we need 5 CRAC units of 227 kW cooling capacity.

At the same time, $P_{\text{cooling,CRAC}}$ should also be equal to the heat generated in the CR:

$$P_{\text{cooling,CRAC}} = Q_{\text{IT}} + P_{\text{CRAC}} \tag{1.23}$$

To be safe, we assume $Q_{\text{IT}} = P_{\text{IT}}$ from here on.

In this way, the coefficient of performance (COP) of the CRAC units will be:

$$COP_{\text{CRAC}} = \frac{P_{\text{cooling, CRAC}}}{P_{\text{CRAC}}} \tag{1.24}$$

which will get to $227kW/6.32kW = 36$ for our typical example in this study.

There is a discussion that if we keep the ΔT high (avoid mixing cold and hot air by isolating their paths or by any other means) the efficiency of the CRACs will increase considerably because of reduction in the required air flow. However, there is a trick here. The sources of heat are usually inside the chassis, and it is highly possible that some hot spots could develop when the air flow is low. Therefore, management of the air flow at local and small scales is very important before any attempt at bigger scales. However, this is out of the scope this study and will be considered in future.

Chillers

The cooling power, or heat handling capacity, of a chiller plant can be expressed as follows:

$$P_{\text{cooling,CH}} = (\mathrm{d}m/\mathrm{d}t)_{\text{CHW}} C_p n_{\text{Chillers}} \left(T_{\text{CHWR}} - T_{\text{CHWS}}\right) \tag{1.25}$$

where T_{CHWR} is the returning water to chillers temperature (return: 14 Celsius degree) and T_{CHWS} is the chiller water source temperature (supply: 7 Celsius degree). The number of chiller units is denoted n_{Chillers}. C_p is 4169 at $25°C$ in (kW/ (m^3/s C°)) and $1/6.817$ in (kW / (cfm F°)). The mass flow $(\mathrm{d}m/\mathrm{d}t)_{\text{CHW,max}}$ can be then calculated as: $1200/(4169 \times 7) = 0.041\text{m}^3\text{/s} = 41\text{kg/s}$.

Also, $P_{\text{cooling,CH}}$ is equal to the heat generated up to the CRAC boundary:

$$P_{\text{cooling,CH}} = P_{\text{IT}} + P_{\text{CRAC}} \tag{1.26}$$

$P_{\text{cooling,CH}}$ has a degradation relation with utilization ratio

$$P_{\text{cooling,CH}} = \frac{P_{\text{cooling,CH,max}}}{P_{\text{chillers,max}}} P_{\text{chillers}} \left(1 - B_{\text{cooling,CH}}(1 - \theta_{\text{Chiller}})^2\right)$$

This is why having multiple but smaller chillers is recommended that also decreases the risk of failure. For the purpose of simplicity, we assume $B_{\text{cooling,CH}} = 0$.

$$P_{\text{cooling,CH}} = P_{\text{cooling,CH,max}} n_{\text{Chiller}} \left(A_{\text{Chiller}} \theta_{\text{Chiller}} + B_{\text{Chiller}} \theta_{\text{Chiller}}^2\right) \tag{1.27}$$

$\text{COP}_{\text{chiller}}$ is the coefficient of performance of a chiller unit: $\text{COP}_{\text{chiller}} = P_{\text{cooling,CH}}/P_{\text{chillers}}$. In market, the $\text{COP}_{\text{chiller}}$ is assumed to be around 5.0 to 3.9 for chiller units (equivalent to 0.7-0.9 kWh/ton). One ton of cooling is equivalent to 12,000 Btu = 3.51685 kWh.

Cooling tower (CT)

The governing equation at the CT are as follows. The heat handling of the water loop is equal to the cooling capacity of the CT:

$$(\mathrm{d}m/\mathrm{d}t)_{\text{CW}} = \frac{P_{\text{cooling, CT}}}{C_p(T_{\text{CWR}} - T_{\text{CWS}})} \tag{1.28}$$

This water mass flow is also related to the utilization ratio of the chiller pumps:

$$(\mathrm{d}m/\mathrm{d}t)_{\text{CW}} = (\mathrm{d}m/\mathrm{d}t)_{\text{CW,max}} \theta_{\text{ChPumps}} \tag{1.29}$$

Note, we assumed $\theta_{\text{ChPumps}} = \theta_{\text{Chiller}}$ in this study. The maximum mas flow $(\mathrm{d}m/\mathrm{d}t)_{\text{CW,max}}$ can be also calculated: $1200\text{MW}/(4169 \times 6) = 0.048\text{m}^3\text{/s} = 48\text{kg/s}$.

At the same time, $P_{\text{cooling,CT}}$ is also equal to the heat generated up to the chiller units' boundary:

$$P_{\text{cooling,CT}} = P_{\text{IT}} + P_{\text{CRAC}} + P_{\text{CH}} \tag{1.30}$$

The cooling capacity of the CT is related to amount of water evaporated:

$$P_{\text{cooling,CT}} = \frac{\mathrm{d}m}{\mathrm{d}t}_{CT} L_p \tag{1.31}$$

where $L_p = 2405$kJ/kg is the latent heat of water evaporation.

For a CT designed to handle 1.2MW heat load (from our typical example of 1MW CR), and a range of 6° and an approach of 5°, we can calculate its performance as follows; its heat capacity is equivalent to a flow of water in the loop with a flow rate of:

$$(\mathrm{d}m/\mathrm{d}t)_{\text{CW,max}} = \frac{1.2\text{MW}}{4169 \text{ J/kg C}^\circ \times 6} = 47.97\text{kg/s} \tag{1.32}$$

The amount of water required to be evaporated at the CT peak power can be calculated as follows:

$$\frac{\mathrm{d}m}{\mathrm{d}t}_{CT,max} = \frac{1.2\text{MW}}{2405 \text{ kJ/kg}} = 0.5\text{kg/s} \tag{1.33}$$

This is equal to $0.5 \times 3600 \times 24 = 43.11$ m^3/day water footprint. In this study, the water footprint is not a target. However, we wanted to point out that the environmental footprint of a datacenter is not limited just to its GhG emissions.

The partial COP of the CT can be estimated as follows:

$$\text{COP}_{\text{CT}} = \frac{P_{\text{cooling,CT}}}{P_{\text{CT}}} \tag{1.34}$$

which will get to $1200kW/48kW = 25$ for our typical example in this study.

Finally, it is important to note that the performance of an air-based cooling system is affected by the altitude of a data center (Fumo *et al.*, 2011)[5]. Considering the geographically-distributed nature of an NDC, the differences in the altitude of the participating data centers would have a big impact on their performance and also consumption.

[5]The altitude affects the T_{wb}, which in turn affects the performance of the cooling system.

1.5 Simulation Platforms for Energy Efficeiny and GhG Footprint in Cloud Computing

In the following sections, several existing simulation platforms are discussed and their main features are presented.

1.5.1 CloudSim

CloudSim is the most famous multi-platform cloud simulation platform, which is developed at the University of Melbourn (Buyya *et al.*, 2009).[6] The mission of the project associated to this simulator is to provide a generalized, and extensible simulation framework that enables seamless modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services. To be more specific, it supports features such as large scale Cloud computing, virtualized server hosts, energy-aware computational resources, specification of data center network topologies and message-passing applications, federated clouds, dynamic insertion of simulation elements, stop and resume of a simulation, user-defined policies for allocation of hosts to virtual machines and policies for allocation of host resources to virtual machines, economic aspects of the cloud market, and dynamic workloads. At the same time, it suffers from the limitations such as absence of GUI, lack of support for geographically-distributed data center configurations, lack of support for geographically distributed user workloads, lack of validation of data and models, and lack of support for variations in network delay due to high demand or equipment failure. Its latest release was CloudSim 3.0.3 on May 2nd, 2013.

In addition to CloudSim, CloudSimEx is a side project to develop extensions for the main CloudSim simulator. For example, MapReduce simulations is available as a feature of CloudSimEx. Furthermore, Cloud Analyst is a tool developed at the University of Melbourne whose goal is to support evaluation of social networks tools according to geographic distribution of users and data centers. This tool characterizes location-aware social-network communities of users and data centers. It calculates parameters such as user experience while using the social network application and also load on the data center. This simulator can be used to perform

[6] http://www.cloudbus.org/cloudsim/

some geographically-distributed scenarios. Its main features are easy to use GUI, ability to define a simulation with a high degree of configurability and flexibility, repeatability of experiments, graphical outputs, and use of consolidated technology and ease of Extension (using Java Swing).

1.5.2 GreenCloud

The lack of detailed simulators on the market was the motivation for University of Luxembourg to develop this Linux-based green cloud simulator that aims at the cloud performance indicators (Kliazovich *et al.*, 2012).[7] GreenCloud is an energy-aware packet-level simulator of cloud computing data centers. It has been elaborated in the context of the GreenIT project with a focus on the communications within a cloud at the packet level.

1.5.3 iCanCloud

iCanCloud is another multi-platform initiative (Núñez *et al.*, 2012),[8] developed at Universidad Carlos III de Madrid, Spain toward simulation of cloud computing data centers. It is written in C++ on the top of SIMCAN Simulation platform which in turn is built on top of OMNeT++ and INET frameworks. SIMCAN was originally developed for simulating HPC systems. The benefit of this architecture is that models of real hardware components were used to construct the underlying core models of iCanCloud. It also provides a GUI and API to generate the distributed models. The main objective of iCanCloud is to predict the trade-offs between cost and performance of a given set of applications executed in a specific hardware, and then provide to users useful information about such costs. In iCanCloud, simulating instance types are provided by Amazon, and therefore their models are included in the simulation framework. Its main features are ability to model and simulate both existing and non-existing cloud computing architectures, providing flexible cloud hypervisor module, providing customizable VMs to quickly simulate uni-core/multi-core systems, providing a user-friendly GUI to ease the generation and customization of large distributed models, providing a POSIX-based API

[7] http://greencloud.gforge.uni.lu/index.html
[8] http://icancloudsim.org/

and an adapted MPI library for modeling and simulating applications, and ability to add new components to the repository of iCanCloud to increase its functionality.

However, it should be noted that in practice it is limited to three virtual machine prototypes. These VMs are Customizable but of fixed resource. For example, the small VM prototype has 1 CPU, 1 disk and 1 GB RAM. Since the iCanCloud simulator is intended to model the Amazon EC2, it was straightforward for its developers to validate it against the actual hardware. An application simulating orbital trajectories of the Mar's moon Phobos was run on both EC2 and iCanCloud. Results for the Cost per Performance benchmark closely matched runs executed on the EC2. Current limitations of iCanCloud are: i) only the Cost per Performance modeling has been validate, and ii) only the EC2 environment has been implemented or tested.

1.5.4 MDCSim

MDCSim is an event driven simulator, integrated on the commercial CSIM (Lim *et al.*, 2009). It works similar to to CloudSim. Below, in Table 1.2, a comparison of these popular cloud simulators is provided. As can be seen from the table, although each one provides some interesting features, none of them are comprehensive in terms of covering both job scheduling on a distributed cloud, and also i) detailed, real-time modeling of the power consumption, ii) real-time inclusion of power mix of the electricity grid, and also iii) detailed modeling of the cooling system are totally absent. These three factors, as we will show in the following chapters, are critical in ecofriendly and green operation of any distributed data center network, and by ignoring them a large margin of error will be imposed on the performance of the system.

Feature	GreenCloud	CloudSim	MDCSim	iCanCloud
Platform	Ns2	Simlava	CSIM	OMNeT/MPI
Language/Script	C++/OTcl	Java	C++/Java	C++
Availability	Open source	Open source	Commercial	Open source
Simulation time	Tens of minutes	Seconds	Seconds	Seconds
Graphical support	Limited (Network animator Nam)	Limited (CloudAnalyst)	None	Yes
Application models	Computation, Data transfer, and Exec. deadline	Computation, Data transfer	Computation	Computation, Data transfer
Communication models	Full	Limited	Limited	INET
Support of TCP/IP	Full	None	None	Full
Physical models	Available using plug in	None	None	Amazon
Energy models	Precise (servers + network)	None	Rough (servers only)	None (Work-in-Progress)
Power saving modes	DVFS, DNS, and both	None	None	None

Table 1.2 Comparison of cloud computing simulators

1.6 Chapter Summary

There is a long history of works on the network of data centers, but they are mainly focused on performance aspects of the system. However, there are some works which are focused on energy topics, but they are not inclusive in regards to some important considerations such as carbon regulations, energy mix variations, energy price variations, temperature variations, global optimization, profit parameters, workload intensity variations, and cooling system variations and optimization.

Some works use DVFS as a tool for energy efficiency. They calculate an optimum frequency for each job which the energy consumption is minimum, but the question remains that how these local optimums can guarantee the global optimum.

The cooling model used in many of these works is simple. However, there are some works on different components of cooling systems, but there is no general model which can be applied to data center with high details. Furthermore, having this detailed model on hand, the cooling system can be optimized for energy efficiency. Also, current models for server metering and server consolidation are not totally accurate, which they can be improved to a higher level of accuracy.

In regards to the web application load balancing, up to our best knowledge there is no study which include data center consolidation in contrast with server consolidation. In addition, a full energy diversity investigation is not conducted on a geographically distributed cloud.

CHAPTER 2

CARBON-PROFIT-AWARE GEO-DISTRIBUTED CLOUD

In the literature review (Chapter 1), a global picture of researches related to energy modeling and efficiency in data centers was provided. As it was summarized in Section 1.6, there are several issues associated with current practical and theoretical solutions which are related to the lack of considerations of carbon regulations, energy mix variations, energy price variations, temperature variations, global optimization, profit parameters, workload intensity variations, and cooling system variations and optimization.

This chapter describes the main idea of this research which is the attempt to fulfill Obj #1 of this thesis. First, it provides details about a state-of-the-art architecture of a Geo-DisC system (baseline). Then, it provides details about improvements to this baseline architecture in many aspects of the system such as objectives, modeling, and algorithms. As is shown in Table 1.1, researchers are considering a variety of different parameters, and it is difficult or infeasible to compare their results individually with other researchers results. Therefore, here, an inclusive architecture is described as the baseline of this research which includes the best practices from various state-of-the-art studies. For example, if research "A" considers the energy price and research "B" considers the energy mix, this baseline considers both energy price and energy mix. Last, result of our proposed architecture will be compared with the result of state-of-the-art algorithms implemented in this baseline architecture.

The first section of this chapter describes the baseline architecture, and the second one addresses the issues discussed in the literature review by suggesting improvement in baseline components or introducing new components such as a new Carbon-Profit-Aware (CPA) scheduler.

2.1 State-of-the-Art Geo-DisC Architecture (Baseline Design)

With the advantage of using cloud computing for resource consolidation and other purposes such as high availability, scalability, and system maintenance, virtual machines are often shuffle

and move to different physical servers to increase energy efficiency. In an organization, this is a common practice nowadays to move applications from underutilized servers to highly utilized servers. Therefore, management of the consolidation of resources is very important in such systems, especially when the servers are geographically distributed. For example, a system like Amazon web applications, has data centers in different locations, and it is possible to run applications in any of these locations with different prices. In Figure 2.1, the schematic of a geographically distributed system is illustrated. As it is shown, in a distributed cloud

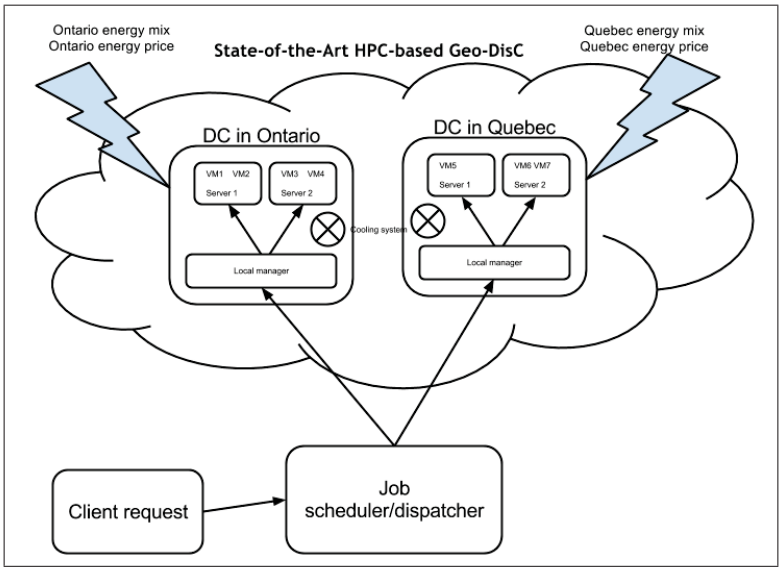


Figure 2.1 Geo-DisC baseline schema

environment, there are several well connected data centers forming one cloud. In fact, Geo-DisC is a big uniform cloud which operate on top of several geographically distributed data centers. As a practice in previous researches, multiple data centers are located at geographically distributed locations to ensure redundancy for business continuity.

These data centers generally consist of computer rooms with many racks of servers. In many larger data centers, there are multiple computer rooms to accommodate large quantities of server racks. To avoid overheat, cooling systems are often required in each data center. By using virtualization technology, as is shown in Figure 2.1, multiple virtual machines can reside in a physical server. These VMs run different type of applications (jobs) in an isolated

environment. A VM can be paused, stopped, saved, resumed, and snapshotted, which are very important in terms of high availability concepts. If a server which is running a job breaks down, that aborted job can be restored in another server with minimum loss of data and service.

Depending on the type of the application (HPC or web), each job needs a certain amount of time and a number of CPU cores to be completed. It is assumed that an HPC application is an application which mainly utilizes CPU core, and the usage of network and disk are negligible. On the other hand, web applications are less CPU demanding, but they may run for longer periods with higher network usage. When a client requests to execute a job, the job scheduler that is located outside of the cloud determines which data center/server and what time interval are best options to handle the job before it dispatches it to the local manager of a specific data center. The job scheduler will determine this suitability based on the job's resource requirements, current workload of each data center, cost of energy source, and greenness of data centers. Once the request reaches the local manager of a specific data center, the manager is responsible to provision the suitable VM on the determined server in the determined time slot.

As each data center may be located at different city, region, or even country, they are bound to a different type of energy suppliers. For instance, a data center residing in Ontario would mainly use nuclear power versus a data center residing in Quebec that would mainly use hydro source. Given that scenario, the type and cost of energy vary for each data center. Besides that, power suppliers often charge a different rate for different time of the day (which also shifted because of their different time zones). These polices includes billing a higher price per kWh during peak hours versus non-peak hours. Moreover, some countries apply carbon tax on fossil-based fuels; such as coal, petroleum, and natural gas to reduce carbon dioxide (CO_2) emissions, the primary cause of global warming.

The main aspects of a Geo-DisC which are data centers, job scheduler, and job trace were described here. More detail on components of this design is provided in the following subsections.

2.1.1 Energy Model

In the previous subsection, a general view of a typical Geo-DisC was presented. In order that job scheduler creates an optimum job schedule, it needs to have access to an accurate estimate of energy consumption of the jobs. Therefore, having accurate energy models are necessary. In previous sections, the energy consumption was mentioned several time. The energy consumption of the system can be achieved in two ways: direct measurement and energy models. Direct measurement is not always possible, and it costs. Specific devices with supported features need to be used in the system in different levels, in order to measure the energy consumption of the components directly. The other problem with direct measuring of the energy is that it is not possible to directly measure the energy consumption of subsystems. In addition, when schedulers and optimizers are producing job plans for the future, they need to estimate the energy consumption of each situation occurring in the future.

For the energy consumption of HPC jobs, the frequency model is used. Which, the power consumption has a relation with the cube of the frequency of the CPU. Bringing down the frequency of CPU will decrease the energy consumption of the server significantly, but the downside is that by doing so, the completion time of the jobs will increase. As mentioned in the literature review, in Garg *et al.* (2011), an optimum frequency for energy consumption is calculated which may not be optimum for the profit of the system. For web applications, the energy model, which is used, is different with HPC jobs and is based on the utilization of the CPUs and servers.

Servers are not the only energy consumers of the system. Cooling systems consume a significant amount of energy which need to be considered in a realistic model. As mentioned in the literature review, a common method for considering the energy consumption of support in a data center including the cooling system is to use the COP or PUE factors (Equations 2.1, 2.2, 2.3, and 2.4).

$$E_{\text{total}} = E_{\text{IT}} + E_{\text{support}} \tag{2.1}$$

$$\text{PUE} = \frac{E_{\text{total}}}{E_{\text{IT}}} \tag{2.2}$$

$$\text{COP} = \frac{E_{\text{IT}}}{E_{\text{support}}} \tag{2.3}$$

$$\text{COP} = \frac{1}{\text{PUE} - 1} \tag{2.4}$$

Here, a global description of energy models is described. These energy models will be used in the following subsection in order to calculate the carbon footprint of the system. A more detailed description of the models is provided in the modeling chapter.

2.1.2 Carbon Footprint and Pricing

In the previous subsection, models were described to measure the energy consumption of systems, but not all the energy sources are similar in terms of being environment-friendly. The goal of this subsection is to describe why calculating the carbon footprint is essential, what parameters are included and how to calculate it.

The role of GhG in global warming and climate change is not hidden for many people, and one of main contributors to GhG is CO_2 (carbon dioxide). Regarding catastrophic phenomenon related to global warming such as sea level rise, it is absolutely necessary to measure and control these dangerous substances.[1]

In this research, the focus is on the carbon footprint, but nevertheless all the non-environment-friendly side-effects of human development are important and need to be considered in more general researches such as Life Cycle Assessment (LCA) studies which is out of scope of this research. Although there is no consistent correlation between the CO_2 footprint and all the other environmental impacts (Laurent *et al.*, 2012), it is safe to assume that some of other negative impacts of a system are also indirectly decreased when the carbon footprint of the system is controlled.

[1]The record holder for the Global Warming Potential (GWP) is Perfluorotributylamine (PFTBA) with a GWP of 7,100 for a 100-year timeframe (Hong *et al.*, 2013). CO_2 is the reference with GWP $= 1$.

When the amount of consumed energy and the type of energy source is known, it is easy to calculate the carbon footprint of the consumed energy by using the carbon emission factor of that particular source of energy. However in the real life, electricity of a grid comes from a mixture of various energy sources, for which an aggregated carbon rate needs to be calculated based on the participating energy sources and the amount of energy they are contributing in the grid.

In this subsection, the main aspects of carbon footprint of a Geo-DisC is discussed which are its contribution to global warming, carbon rate, and energy mix. Models described here are used to calculate the energy consumption and carbon footprint of a typical Geo-DisC, but in order to calculate the total profit of the system, a model should be used to estimate the total cost of the system based on its consumed energy and carbon footprint. In most work, a flat rate is used to calculate the cost of energy. In the following subsections, the scheduling component will be described.

2.1.3 Scheduler Features

Some metrics were described in the previous sections as the parameters of the system. A scheduler can use this information to schedule a trace of jobs on a Geo-DisC system. The main goal of a scheduler is to shovel around the jobs in order to reach its defined goal(s) while respecting jobs restrictions. The goal of a scheduler can be to minimize the completion time of jobs or minimize the number of failed jobs or minimize the energy consumption or carbon footprint of the jobs or maximize the profit or achieve some of these goals all together.

Usually schedulers use a greedy approach for achieving one goal and multi-level approaches for multi-objective scenarios. For example if the goals of a scheduler are to minimize the carbon footprint and maximize the profit, it may sort the possible actions based on one goal and sort the result of the first sort based on the other goal in order to satisfy both goals.

The traditional scheduling is done by schedulers such as Min-Min Completion Time, and the newer environment-friendly ones use schedulers such as MIN-Min Carbon Emission in order to minimize the carbon while maximizing the profit, but it has some problems as mentioned

in the literature review. If the utilization is 100%, the algorithms do not work as they should work. This claim is proven in the experimental results section.

The schedulers may also control the frequency of CPUs which are running the jobs in order to achieve their goals. Usually this is done by calculating an optimum frequency for the jobs which has the lowest energy consumption.

The main aspects of a scheduler are having accurate measures related to the goals of the scheduler, an algorithm which is able to achieve multiple goals at the same time, and a strategy for adjusting the frequency of the CPUs. In the following subsection, another aspect of a Geo-DisC will be discussed which is workload features.

2.1.4 HPC Workload Features

In the previous subsections, most of the components of a Geo-DisC were described. In this subsection, the main features of the workload of such system will be explained. In order to have realistic results, HPC traces are used from recorded real systems. For special use cases, HPC traces are filtered to represent a workload with a specific feature or features. It is very important to recognize the amount of load of jobs which is utilizing the system. Some algorithms are good with certain utilization percentage of the system, and may not perform well when the system is 100% utilized.

Based on the entry time of the jobs, their length and deadline of the jobs, number of needed CPU cores, and also the type of scheduling algorithm, some jobs being scheduled, and some failed which have a direct effect on the quality of service of the system.

In summary, based on the type of jobs and amount of utilization of the system, performance of scheduling algorithms may differ which needs to be considered in the comparison of different algorithms.

2.1.5 Summary

In this section, the common practice architecture of state-of-the-art research on HPC job scheduling is presented which is a network of data centers forming a uniform cloud. Also, models to calculate the energy consumption of a typical system is described. Based on the energy consumption and energy mix of the region, it is possible to calculate the carbon footprint of the system. With having the energy consumption, carbon footprint, and operational cost of the system, the total cost of the system can be calculated, therefore, the profit of the system is measureable.

A scheduler will act on the measures like energy consumption, carbon footprint, profit, and job features to achieve its multiple goals such as maximizing the profit or minimizing the carbon footprint or increasing the Quality of Service or Quality of Experience of the customers. The whole system is a uniform cloud to take advantage of cloud features such as server consolidation and high availability. Job features such as system utilization percentage has a direct effect on the performance of the schedulers which need to be considered in the experimental comparisons. A system design able to address issues related to the baseline will be described in the following section.

2.2 Carbon-Profit-Aware Geo-DisC Architecture (Our Proposed Design)

In the previous section, a baseline for a network of data centers is defined to process HPC jobs based on the state-of-the-art researches. In the introduction, literature review and the previous section (Section 2.1), the issues related to components of such system are mentioned and discussed. Here, to address those issues, a new system, which is a series of improvements to the baseline system, is introduced as a whole and its components are described. More details on each component are provided in the following sections. By adding new components to the baseline model, the new design should have better performance in reducing the cost and carbon footprint of the system at the same time. It is expected that the new design be more realistic than the baseline design since some components like cooling system are modeled with a high degree of details. The schema of the new system is presented in the Figure 2.2.

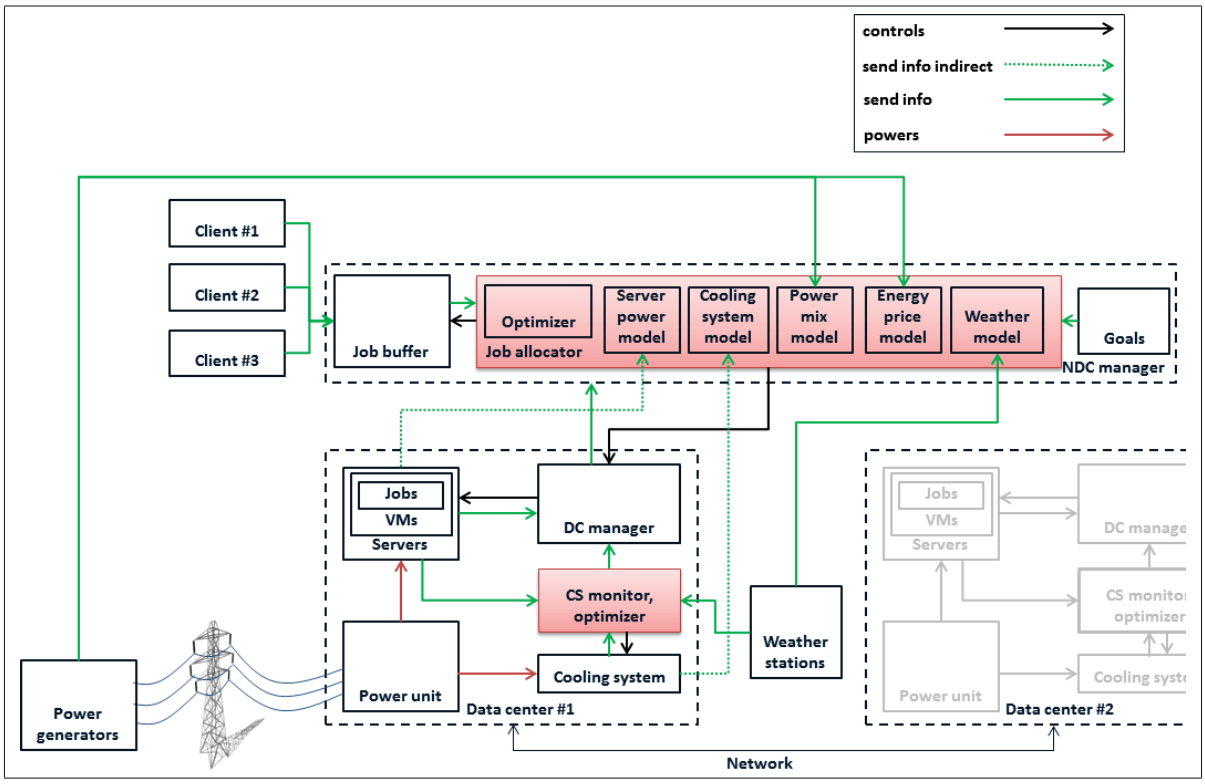


Figure 2.2 Carbon-Profit-Aware Geo-DisC schema

In this new design, several well-connected data centers, which are all supporting virtualization technology, are considered while seamless VM migration between them is available. In each data center, a power unit powers the servers and the cooling system. The cooling system is controlled and optimized by a local optimizer. The local cooling system optimizer gets information from servers and local weather station. These information provide the local optimizer with how much heat is produced in the data center and how much is the outside temperature. With these information, the local optimizer can adjust the performance of the cooling system efficiently. All the information regarding the servers and cooling system is transferred to the Network of Date Centers (NDC) manager through the DC managers. In the NDC manager, there are models for every piece of the information in the system. The job allocator uses these models and information for scheduling or load balancing of the jobs are being received in the job buffer. The jobs in the job buffer are coming from various clients with different requirements. The job allocator uses the DC managers to schedule and load balance the jobs on the end servers.

In the following subsections, first the modeling of the new design will be discussed, then a full description of the new scheduler will be provided. Then the concept of load balancer will be discussed, and finally the role of different managers and controllers in the system will be discussed.

2.2.1 Component Modeling

Modeling plays an extremely prominent role in this research. Without accurate and complete information, producing a good decision seems impossible.

The performance of CPA scheduler is directly related to the accuracy of the governing models. In this research, a detailed model for energy consumption of data centers is provided including servers and cooling systems.

In addition to the calculation of energy and carbon footprint of the system which are described in the previous section, a new measure is introduced in this research as greenness of the system for modules or actions, which is a number between 0 to 1, which shows how much a data center or an NDC or a server is green compare to the dirtiest available source of energy (Equation 3.4).

In addition, to have a realistic measure of energy consumption of the data centers, a very detailed model for energy consumption of cooling systems is introduced in this research.

In most work, a flat rate is used to calculate the cost of energy for services. However, in many places of the world, there are different rates for energy consumption in different hours of a day and different seasons. Accessing to the price chart of a region, enable the scheduler to estimate the price of energy in the future while making schedules for coming hours. In some places, it is possible to buy the energy from an energy market with possibly lower rates, but considering those energy markets are out of scope of this research.

In addition to the price of energy in some states, there is already a carbon tax in place which will add another cost for carbon footprint to the total cost of the service. There is also other related operational costs such as building rental, personnel, hardware/software investment, and

network which need to be added to the total cost of the services. In addition, corporation tax needs to be considered in order to calculate the net profit of the services.

With considering the energy (IT plus support, i.e. cooling) cost, carbon tax, operational cost, and corporation tax, it is possible to estimate the total cost of the whole system or a single service. The profit of the system can be calculated easily by subtracting the total cost from the revenue. In the next subsection, the main elements of CPA scheduler will be discussed.

2.2.2 Carbon-Profit-Aware Scheduler

In the previous section, new modeled metrics of the new design were described. A new scheduler can use these new metrics to achieve new goals. Carbon-Profit-Aware scheduler is a scheduler which is aware of many parameters of the system.

In CPA scheduler, the frequency of CPUs are calculated in such a way that the objective of the system is satisfied. There are many parameters which may contribute in lower profit of a Geo-DisC such as increase in electricity price, increase in environment temperature, and decrease in greenness of the energy mix. A good scheduler is aware of all these parameters, and because the scheduler assigns the jobs to the core-time slots ahead of the time, it is highly important for a CPA scheduler to have access to the future changes of the rates or predict the weather related parameters.

The main module of the new scheduler is the optimum frequency calculator. The scheduler uses this component to optimum the Geo-DisC towards its goals. This component is described in the Section 4.3.1.

2.2.3 MLGGA Load Balancer for Web Applications

Previous section describes the job scheduler for HPC jobs, however, for web applications, another type of controller needs to be used. Since main job scheduler estimates the efficiency of a job based on energy prices and energy mix, its performance will be low if the length of a job is beyond planned scope of the scheduler. For this type of job, which are mainly

web applications, another component is required to dynamically migrate the jobs for efficient utilization and GhG emissions reduction of system. In the Chapter 5 details, description of MLGGA algorithm are explained.

2.2.4 Managers and Controllers

There are several controllers and managers in this design, which together enable the system to achieve its goals. In Figure 2.3, different modules of the system are represented in a stacked graph. As it is shown, the job controller module, schedule the jobs on VMs. It simply receives the job schedule from CPA job scheduler, and it is able to create/delete VMs from servers through hypervisors. Higher level modules such as NDC monitor and manager have access to a wider range of data from their underlying modules, and they are able to control those modules.

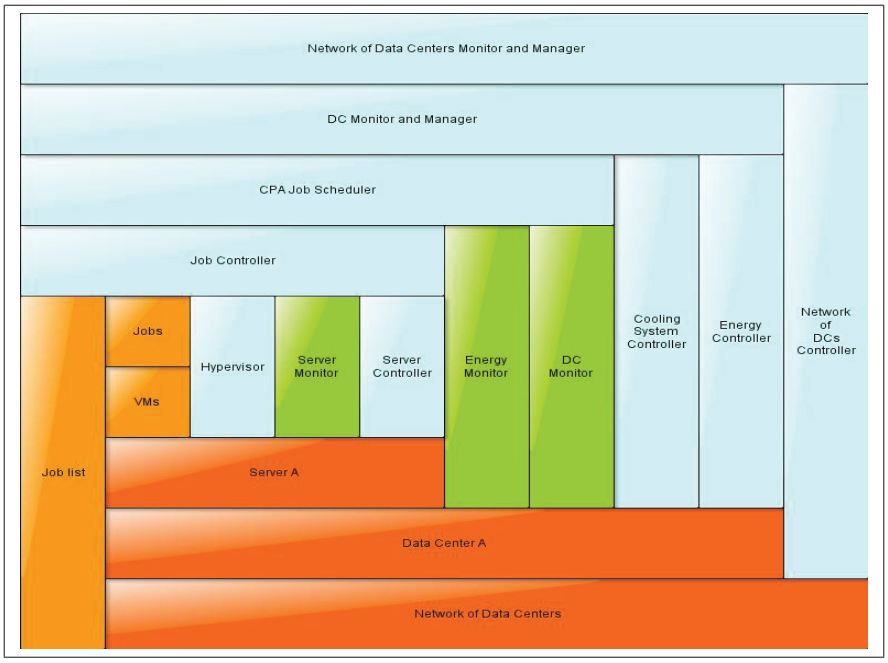


Figure 2.3 Carbon-profit-aware Geo-DisC stacked graph

NDC Manager is on top of the whole system, and any other subsystem is managed directly or indirectly by this component. This component is responsible for running the whole system

and achieving its goals. A DC manager acts like the NDC manager but in smaller scale. It changes the energy and cooling system parameter of a data center by using the energy and cooling system controller modules.

Figure 2.4 shows the flow of actions between managers and controllers. First, NDC manager receives the requests from clients and updates the Jobs list. Next, CPA scheduler create a schedule for the requested jobs on the available servers. It receives the necessary information about the servers, energy and cooling system from job controller and NDC manager, respectively. Then, the job controller execute the job schedule plan on the servers through the hypervisors. It also has control on the servers through server controller such as the capability to turn on/off, to put on standby, and to adjust the CPU frequency. Server monitor report the status of the server to the job controller such as utilization and PMC metrics. On the other hand, NDC manager controls the cooling system and energy parameters of the system through the DC managers, and energy and cooling controller.

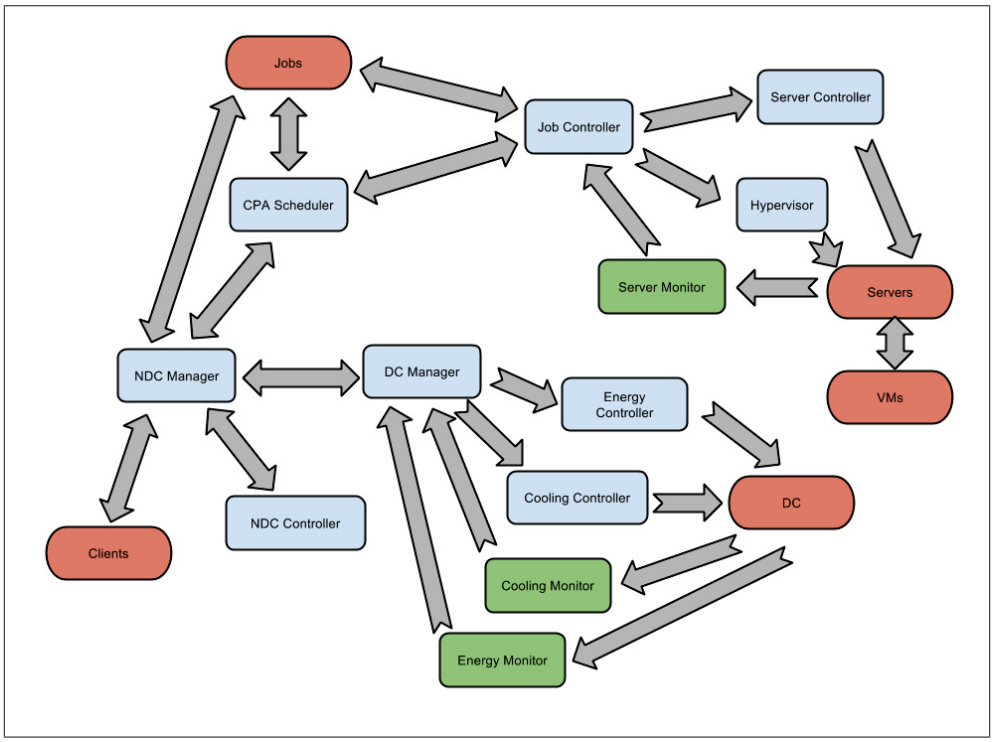


Figure 2.4 Carbon-profit-aware Geo-DisC control cycle

2.2.5 Summary

In this section, components of an improved design for Carbon-Profit-Aware Geo-Distributed Cloud are described. This design can be used for HPC jobs by using a CPA scheduler or web applications by using an MLGGA load balancer. The goal is to increase profit while minimizing the carbon footprint of the whole system. In order to have a realistic profit model, several components are modeled which have a direct impact on the cost of the whole system such as cooling systems, energy price, and tax.

2.3 Chapter Summary

In this chapter, a baseline system for Carbon-Profit-Aware Geo-Distributed Cloud is introduced based on best practices from state-of-the-art studies. Main components of such system are a multi-objective scheduler and energy and carbon models. Also, description of a new CPA Geo-DisC is described with new models and new scheduler and load balancer.

Components of the new design are described in details in the following chapters, and the new design is compared with the baseline by carrying out several experiments, and comparison results are provided in the final chapter.

CHAPTER 3

GEOGRAPHICALLY DISTRIBUTED CLOUD MODELING

As it was mentioned in the introduction chapter, to have accurate results, accurate models are needed such as energy consumption model of cooling systems. In this chapter newly introduced models for cooling system, server, and profit-per-service are presented. Some models are used in the HPC job scheduler, and some are used in the web application load balancer which are explained in the following chapters. There are two main elements in a data center which are responsible for the power consumption of a data center: the IT equipments and the cooling system. In the following sections, first, a model is presented to estimate the power consumption of the servers in Section 3.1.2, and then a model is presented and optimized for power consumption of the cooling system in Section 3.2. Having the power consumption of the servers and cooling system, it is possible to calculate the total power consumption of the data center. If the power mix of the region is known, it is easy to calculate the carbon footprint of the system based on the power consumption of the data center. Considering other parameters of a data center such as electricity price, carbon tax, sales rate, and other taxes, it is possible to calculate the profit of the data center (Section 3.1.1). Last, in order to evaluate the greenness of the data centers some new meterics are defined in the Section 3.1.3

3.1 IT Equipment Modeling

In this section, several models will be introduced for calculating the necessary measurements in a data center. Since in this research, our main objective is maximizing the profit of data centers, in Section 3.1.1 a model is presented for calculating the profit of a load unit of the data center. Later, this model will be used for calculating the total profit of the data centers in the definition of the optimization problem, and it will be also used in the scheduler as a guide. In order to calculate the carbon footprint of the data centers, it is necessary to calculate the energy consumption of them. Therefore, in Section 3.1.2, a model is presented to calculate the power consumption of the underutilized servers. As mentioned before, the servers are not the only power consumers of the data centers and cooling system has a significant power consumption

which needs to be considered in the calculation of the power consumption of the data centers. Hence, in Section 3.2, a model is introduced to calculate and optimize the power consumption of the cooling system.

3.1.1 Profit per Core-Hour-GHz

In Chapter 4, a new measure will be introduced as Profit Per Core-Hour-GHz (PpCHG) which represent the amount of profit of the system associated with running an HPC job on one CPU core for one hour while the CPU frequency is f. PpCHG is the main characteristics of the CPA algorithm. In Equation 3.1, a formula for PpCHG is presented, where $\text{TAXR}_{\text{corp}_{\text{region,effective}}}$ $(\%)$ represent the effective corporation tax rate of the business in the previous year in the region where the business files the tax, $\text{SR}_{\text{core,hour,gig}}$ (USD) represents the sales price for a CPU core running for an hour with CPU frequency of 1 GHz, f_{cpu} represents the frequency of the CPU, β_{cpu} and α_{cpu} represent the energy model parameters of CPU, $\text{EP}_{\text{region}}$ (USD) represents the energy price rate of the region, $g_d(t)$ $(\%)$ represents the greenness of the data center, ρ_{max} represents the energy-carbon conversion rate (emission factor, kgCO_2 per kWh) for the dirtiest source of energy (coal), $\text{CTR}_{\text{region}}$ (USD per kgCO_2) represents the carbon tax rate of the region, $E_{\text{cooling}_{\text{core-hour}_{\text{average}}}}$ (kWh) represents the energy consumption of the cooling system in one hour divided per number of CPU cores in the system, $\text{OPEX}_{\text{core,hour}}$ (USD) represents the operational cost of the data center associated with the running a core of CPU for an hour, and $\text{TAXR}_{\text{sales}_{\text{region}}}$ $(\%)$ represents the sales tax rate of the region. In this research the working frequency of the CPU will be the variable of the equation and the other parameters are assumed to be known or if they are not known it is assumed that they are predicted with the best knowledge available.

$$
\begin{aligned}
\text{PpCHG} &= (1 - \text{TAXR}_{\text{corp}_{\text{region,effective}}}) \Bigg(\text{SR}_{\text{core,hour,gig}} * f_{\text{cpu}} \\
&- (\beta_{\text{cpu}} + \alpha_{\text{cpu}} f_{\text{cpu}}^3) * \text{EP}_{\text{region}} \\
&- (1 - g_d(t)) * \rho_{\text{max}} * (\beta_{\text{cpu}} + \alpha_{\text{cpu}} f_{\text{cpu}}^3) * \text{CTR}_{\text{region}} \\
&- E_{\text{cooling}_{\text{core-hour}_{\text{average}}}} * \text{EP}_{\text{region}} \\
&- (1 - g_d(t)) * \rho_{\text{max}} * E_{\text{cooling}_{\text{core-hour}_{\text{average}}}} * \text{CTR}_{\text{region}} \\
&- \text{SR}_{\text{core,hour,Gig}} * f_{\text{cpu}} * \text{TAXR}_{\text{sales}_{\text{region}}} - \text{OPEX}_{\text{core,hour}} \Bigg)
\end{aligned}
\tag{3.1}
$$

The first term of the equation represents the corporation tax and amount of money produced by the sales rate. The second term of the equation represents the energy cost of energy consumed in the servers. The third term of the equation represents the carbon cost of emission produced in the servers. Similarly, forth and fifth terms of the equation represent the energy and carbon cost associated with the cooling system. The sixth term of the equation represents the cost of sales tax and the last term of the equation represents the operational costs of the running a core of CPU for one hour.

Since in this thesis, we only consider the cooling system as the main energy consumer of support system, then $E_{\text{cooling}} = E_{\text{support}}$. According to Equations (2.3) and (2.4), the E_{support} can be rewritten in the form of $E_{\text{support}} = (\text{PUE} - 1) E_{\text{IT}}$. Therefore in this equation the $E_{\text{cooling}_{\text{core-hour}_{\text{average}}}}$ can be substitute with $E_{\text{IT}_{\text{core-hour}}} * (\text{PUE}_{\text{average}} - 1)$, where $E_{\text{IT}_{\text{core-hour}}} = (\beta_{\text{cpu}} + \alpha_{\text{cpu}} f_{\text{cpu}}^3)$. Considering the $E_{\text{cooling}_{\text{core-hour}_{\text{average}}}} = (\beta_{\text{cpu}} + \alpha_{\text{cpu}} f_{\text{cpu}}^3) * (\text{PUE}_{\text{average}} - 1)$, Equation (3.1) can be rewritten as Equation (3.2).

$$
\begin{aligned}
\text{PpCHG} &= (1 - \text{TAXR}_{\text{corp}_{\text{region,effective}}}) \Bigg(\text{SR}_{\text{core,hour,gig}} * f_{\text{cpu}} \\
&- (\beta_{\text{cpu}} + \alpha_{\text{cpu}} f_{\text{cpu}}^3) * \text{PUE}_{\text{average}} * \text{EP}_{\text{region}} \\
&- (1 - g_d(t)) * \rho_{\text{max}} * (\beta_{\text{cpu}} + \alpha_{\text{cpu}} f_{\text{cpu}}^3) * \text{PUE}_{\text{average}} * \text{CTR}_{\text{region}} \\
&- \text{SR}_{\text{core,hour,Gig}} * f_{\text{cpu}} * \text{TAXR}_{\text{sales}_{\text{region}}} - \text{OPEX}_{\text{core,hour}} \Bigg)
\end{aligned}
\tag{3.2}
$$

3.1.2 Power Metering Model for Servers

In Farrahi Moghaddam *et al.* (2011), we combine the models mentioned in literature review (Section 1.3) in order to include all the types of energy consumption involved in a distributed cloud. We also introduce what we call a "greenness" factor for each data center, which enables conversion from energy consumption to carbon footprint in a data center, as follows:

$$C_{p_d}(t) = \rho_d(t) P_d(t) \tag{3.3}$$

$$\rho_d(t) = (1 - g_d(t))\rho_{\max} \tag{3.4}$$

where $C_{p_d}(t)$ represents the carbon footprint per unit time of a data center, $P_d(t)$ represents the power consumption of a data center, and $\rho_d(t)$ represents the carbon emission factor for that particular data center. $\rho_{\max}$ represents the carbon emission factor for the dirtiest source of energy (0.9 kg per kWh Lenzen (2010)), $g_d(t)$ represents the greenness factor of the data center ($g_d(t) = 0$ means 0% clean data center, and $g_d(t) = 1$ means 100% clean and green data center at time t).

An improved version of the initial model for the carbon footprint per unit time is introduced in (Farrahi Moghaddam *et al.*, 2012b), as follows:

$$\begin{aligned} C(t, \Delta t) = \; & C^{(S)}(t, \Delta t) + C^{(N)}(t, \Delta t) \\ + \; & C^{(R)}(t, \Delta t) + C^{(M)}(t, \Delta t) \\ + \; & C^{(U)}(t, \Delta t) + C^{(O)}(t, \Delta t) \end{aligned} \tag{3.5}$$

where $C(t, \Delta t)$ is the total carbon footprint of the distributed cloud from time t to $t + \Delta t$:

$C^{(S)}(t, \Delta t)$, $C^{(N)}(t, \Delta t)$, $C^{(R)}(t, \Delta t)$, $C^{(M)}(t, \Delta t)$, $C^{(U)}(t, \Delta t)$, and $C^{(O)}(t, \Delta t)$ represent the portion of the carbon footprint related to the servers, network devices, storage devices, migration of VMs among servers, cooling, Power Distribution Unit (PDU), and lighting utilities, and turning servers and other electrical devices on or off in the data centers, respectively. For

calculating the carbon footprint of migrations ($C^{(M)}(t, \Delta t)$), the following equation is used:

$$
\begin{aligned}
C^{(M)}(t, \Delta t) = \; & \textstyle\sum_{m \in \mathbb{M}_{\Delta t}} \rho_{\max} \big\{ \\
& r_{d_{a_m}}(t_m) \big\{ \Delta p^{(S)}_{a_m}(t_m) + \\
& \textstyle\sum_{j \in \mathbb{N}_{d_{a_m}}} O^{(N)}_j \Delta p^{(N)}_j(t_m) \big\} \\
& + r_{d_{b_m}}(t_m) \big\{ \Delta p^{(S)}_{b_m}(t_m) + \\
& \textstyle\sum_{j \in \mathbb{N}_{d_{b_m}}} O^{(N)}_j \Delta p^{(N)}_j(t_m) \big\} \\
& \big\} \Delta m_t
\end{aligned}
\tag{3.6}
$$

where $\mathbb{M}_{\Delta t}$ represent the set of migrations which happen in Δt. a_m and b_m represent migration source and destination servers. t_m is the time when the migration m took place. $\Delta p^{(S)}_{a_m}(t_m)$ represent extra power consumption of server a_m during the migration time Δm_t. $\Delta p^{(N)}_j(t_m)$ represent extra power consumption of network elements. These models need to be improved and validated.

In this section, we introduce a new metering model to integrate and to improve the previous models. In Kansal *et al.* (2010), a linear function was fitted on server resource usage. In parallel, in Bertran *et al.* (2010b), a linear function was fitted on server PMC counters, and in Farrahi Moghaddam *et al.* (2012b), a linear function was fitted on both server resource usage and PMC counters. Here, we propose that a Piecewise-Linear Regression (PLR) on both server resource usage and PMC counters can more accurately provision the energy consumption of a server, as follows:

$$
p(t) = r(\text{pmc}_1, \text{pmc}_2, \cdots, \text{pmc}_n, c, d, n, m) \tag{3.7}
$$

where $p(t)$ represents the consumed energy of a server per hour. pmc_i, c, d, n, and m represent i^{th} PMC counter, CPU, disk, network, and memory utilization, respectively. r represent the PLR function.

We use the Adaptive Regression Splines (ARESLab) toolbox (Jekabsons, 2011) in order to perform piecewise-linear regression of the server energy consumption using CPU, memory, disk, and network usage and PMC counters as independent variables. A 10-fold cross validation is used to calculate the accuracy of the model. This model then is used to estimate the energy

consumption of the servers, processes if their PMC counters and resource usage are known. In section 6.3, the proposed model is evaluated on real servers and the energy prediction results are compared with results obtained using other models from Kansal *et al.* (2010), Bertran *et al.* (2010b), and Farrahi Moghaddam *et al.* (2012b).

3.1.3 NDC Carbon-Related Metrics

Sensitivity to Intermittent Sources of Energy of Network of Data Centers

As mentioned in the introduction, it is important to know how robust the design is and how weather condition fluctuations can affect the performance of the system. Therefore, we examine our design under different weather conditions and analyze the results to provision the sensitivity of our design and our consolidation algorithm to weather conditions.

To calculate the sensitivity of a design to any weather conditions, a plane, $y = b - a_s s - a_w w$, is fitted to the carbon measurements under different weather conditions (using Matlab least squares linear regression), where y is the measured carbon footprint, s is the percentage of solar energy, and w is the percentage of wind energy; and a_s and a_w represent the sensitivity of each scenario to the solar energy percentage and wind energy percentage in kilograms of CO_2 (kgC) respectively.

Sensitivity to the solar or wind energy percentage refers to how much additional carbon footprint will be added to the total carbon footprint if the solar or wind energy percentage decreases from 100% to 0%. Smaller sensitivity values for a scenario show more stability, and less risk of consuming greater amounts of non green energy in bad weather conditions.

Greenness Factor of Network of Data Centers

In this section, we introduce another metric for the evaluation of a design which is the "CAD-Cloud greenness factor," G, which is calculated for each time period as $G = 1 - C/C_0(E)$, where C represents the carbon footprint, and $C_0(E)$ represents the carbon footprint equivalent to the total consumed energy of the scenario produced by the dirtiest source of energy (coal),

(Farrahi Moghaddam *et al.*, 2011). A CADCloud greenness factor of $G = 100\%$ represents a zero carbon CADCloud. Note that the G factor is different from the factor $g_d(t)$ used in Equation (3.4), which is the greenness factor of a data center. In contrast, G is the greenness factor of the whole CADCloud, which is powered by various sources of energy.

3.2 Cooling System Modeling

3.2.1 The Temperature Altitude Aware Model (TAAM)

We assume that the cooling system of the data center is the traditional air/liquid/air cooling system. As shown in Figure 1.4, the CRAC units provide a cycle of cold/hot air inside the data center that goes through the chassis and racks. The cold air absorbs the heat generated by IT equipments (and also other sources of heat, such as i) the lighting, ii) solar heat (if there is a window), iii) humidifiers (using evaporative methods), iv) CRACs themselves (the electrical energy used in their fans will be converted into heat when the moving air slows down)). It is usually assumed that %98 of the IT consumed power is converted into heat (Sawyer, 2004): $Q_{\text{in}} = 0.98 P_{\text{IT}}$. However, in this work, we assume all the IT consumed power is converted into heat. This heat introduces a $\Delta T = T_{\text{CR}} - T_{\text{CHWS}}$ increase in the temperature of circulating air in the computer room.[1] In CRACs' cooling coils, the heat is transferred to the cooling liquid (water) in a similar way. The cold water is provided by the chillers, and the heat exchange introduces an increase in the water temperature. Finally, cooling tower allows heat exchange with the outside air. Usually, because of bigger mass of the outside air compared to the cooling liquid mass, a micro-canonical assumption can be made, and it can be assumed that the temperature of atmosphere stays constant.

The majority of governing equations and models of each sub-part of the cooling system, i.e., the CRAC, the chiller plant, and the cooling tower, have been discussed and presented in section 1.4. These models cover both aspects of power consumption and also the cooling capacity of these sub-parts.

[1] CHW stands for Chiller-water, and CHWS denotes source chiller-water that enters CRAC.

In addition, a coefficient of performance (COP) is defined for the chiller plant and also for the cooling tower. The COP of chiller units is defined as the ratio of heat handled by them divided by their energy consumption:

$$\text{COP}_{\text{CH}} = \frac{P_{\text{cooling,CH}}}{P_{\text{CH}}} \tag{3.8}$$

Therefore, this value has a inverse relation with the PUE value: $COP = 1/(\text{PUE} - 1)$, where PUE is the partial (ignoring the rest of support equipments) PUE of the chiller. A typical value of the COP for a chiller plant is around 4; the typical cooling cost (electricity consumption per ton of cooling) of a chiller system is around 0.7 Kw/ton to 0.9 Kw/ton, where one ton of cooling is defined as 12,000 Btu = 3.51685 kW (Energy Design Resources, 2010). Therefore, the partial (ignoring the rest of support equipments) PUE of chiller units can be estimated as: PUE $= (0.8 + 3.51685)/(3.51685) = 1.23$, for 0.8 kW/ton, which in turn results in a COP of $4.35 (= 1/(\text{PUE} - 1))$. We choose $P_{\text{chiller,max}} = 80kW$ for each chiller assuming a chiller coefficient of performance (COP) of 4 with $n_{\text{Chiller}} = 5$ for a 1MW CR in this study.

Furthermore, for a chiller plant, the *range* is defined as the difference in temperature between entering and exiting water in/from chiller condensers in the CRAC water loop that is $T_{\text{CHWR}} - T_{\text{CHWS}}$,[2] and the *approach* is defined as the difference in temperature between exiting water of CRAC loop and entering water of CT loop: $T_{\text{CHWS}} - T_{\text{CWS}}$.[3] In our typical datancenter, we choose a chiller range of 7° and a chiller approach of 6°.

For our typical example of $n_{\text{Chiller}} = 5$ and with assuming $\theta_{\text{ChPump}} = 0.5$, we have

$$P_{\text{ChPumps}} = 0.3{\times}80\text{kW}{\times}5{\times}\left(0.338249 \times 0.5 + 0.972488 \times (0.5)^2 - 0.291451 \times (0.5)^3\right) = 45\text{kW}$$

as the power consumption of all 5 pumps. The power consumption of secondary pumps that move the cold water in the CRAC-chiller loop are assumed to be included in the chillers' pumps consumption for the purpose of simplicity.

The same notation is used for the cooling tower; for a CT, the *range* is defined as the difference in temperature between entering and exiting water in/from CT in the water loop that is $T_{\text{CWR}} -$

[2] CHWR stands for return chiller-water that exits CRAC.

[3] CW stands for cooling tower-water, and CWS denotes source cooling tower-water that enters chillers.

T_{CWS},[4] and the *approach* is defined as the difference in temperature between exiting water and wet blub temperature of outdoor air: $T_{\text{CWS}} - T_{\text{wb}}$. In our typical datancenter, we choose a CT range of 6° and a CT approach of 5°. The *effectiveness* of a CT is defined as:

$$\mu_{\text{CT}} = \frac{T_{\text{CWR}} - T_{\text{CWS}}}{T_{\text{CWS}} - T_{\text{wb}}} \tag{3.9}$$

It is assumed that μ_{CT} is design-dependent, and only changes with θ_{CT}:

$$\mu_{\text{CT}} = \mu_{\text{CT,max}}\theta_{\text{CT}} \tag{3.10}$$

For our typical example, $\mu_{\text{CT,max}} = 6/5 = 1.2$.

These relations along with those presented in section 1.4 form our model of the cooling system as is discussed in the next section.

3.2.2 Set of Equations of the Cooling System Model

In terms of the cooling system, there are 25 (=22+3) variables in the picture:

- Three of these variables are taken from measured weather information: outdoor (dry bulb) temperature T, relative humidity φ and outdoor air pressure p.
- One variable is a direct function of the above three variables: wet-bulb temperature of outdoor air T_{wb} (Fumo *et al.*, 2011).
- One variable is the target temperature in the CR: T_{CR}. We assume this temperature is fixed in all datacenters and fixed to a goal value of $25^{\circ}C$.
- *(before CRAC boundary)*: Four variables are related to CRAC units: θ_{CRAC}, P_{CRAC}, $P_{\text{cooling,CRAC}}$, and ϕ_{CRAC}. We assumed that all CRAC units in a specific datacenter work at the same utilization ratio for purpose of simplicity.

[4] CWR stands for return cooling tower-water that exits chillers.

- *(before Chiller units boundary)*: Nine variables are related to chiller plants: T_{CHWS}, T_{CHWR}, $(\mathrm{d}m/\mathrm{d}t)_{\text{CHW}}$, θ_{CH}, $P_{\text{cooling,CH}}$, P_{CH}, P_{Chillers}, P_{ChPumps}, and θ_{ChPumps}.
- *(before CT boundary)*: Finally, seven variables are related to the CT: T_{CWS}, T_{CWR}, $(\mathrm{d}m/\mathrm{d}t)_{\text{CW}}$, μ_{CT}, θ_{CT}, $P_{\text{cooling,CT}}$, and P_{CT}.

In parallel, we have 17 governing equations (in sections 3.2.1 and 1.4). The equations are as follows:

- One equation to calculate T_{wb} (Fumo *et al.*, 2011).
- There are four equations at CRAC units: (1.9), (1.10), (1.22), and (1.23).
- There are seven equations at chiller units: (1.25), (1.26), (1.11), (1.13), (1.14), (1.15), and (1.27).
- There are five equations at cooling tower: (1.28), (1.30), (3.9), (3.10), and (1.18).

The system of equations are summarized in the Equation 3.13.

$$
\begin{cases}
T_{wb} \quad \text{(Fumo et al., 2011)}, \\
P_{\text{CRAC}} = P_{\text{CRAC,fan}} = P_{\text{CRAC,fan,max}}\theta_{\text{CRAC}}^{2.75}, \\
\phi_{\text{CRAC}} = \phi_{\text{CRAC,max}}\theta_{\text{CRAC}}, \\
P_{\text{cooling,CRAC}} = A_{\text{CRAC,cooling}}\phi_{\text{CRAC}}n_{\text{CRAC}}\left(T_{\text{CR}} - T_{\text{CHWS}}\right), \\
P_{\text{cooling,CRAC}} = Q_{\text{IT}} + P_{\text{CRAC}}, \\
P_{\text{CH}} = P_{\text{chillers}} + P_{\text{ChPumps}} + P_{\text{SecPumps}}, \\
P_{\text{chillers}} = P_{\text{chiller,max}}n_{\text{Chiller}}\left(A_{\text{Chiller}}\theta_{\text{Chiller}} + B_{\text{Chiller}}\theta_{\text{Chiller}}^2\right), \\
P_{\text{ChPumps}} = PCRP_{\text{chiller,max}}n_{\text{ChPumps}}\times \\
\qquad \left(A_{\text{ChPump}}(\theta_{\text{ChPump}}) + B_{\text{ChPump}}(\theta_{\text{ChPump}})^2 - C_{\text{ChPump}}(\theta_{\text{ChPump}})^3\right), \\
\theta_{\text{ChPump}} \simeq \theta_{\text{Chiller}}, \\
P_{\text{cooling,CH}} = (\mathrm{d}m/\mathrm{d}t)_{\text{CHW}}C_p n_{\text{Chillers}}\left(T_{\text{CHWR}} - T_{\text{CHWS}}\right), \\
P_{\text{cooling,CH}} = P_{\text{IT}} + P_{\text{CRAC}}, \\
P_{\text{cooling,CH}} = \frac{P_{\text{cooling,CH,max}}}{P_{\text{chillers,max}}}P_{\text{chillers}}\left(1 - B_{\text{cooling,CH}}(1-\theta_{\text{Chiller}})^2\right), \\
P_{\text{CT}} = \text{CCR}\ P_{\text{chiller,max}}n_{\text{Chiller}}\theta_{\text{CT}}^{2.75} = P_{\text{CT,max}}\theta_{\text{CT}}^{2.75}, \\
(\mathrm{d}m/\mathrm{d}t)_{\text{CW}} = \frac{P_{\text{cooling, CT}}}{C_p(T_{\text{CWR}} - T_{\text{CWS}})}, \\
P_{\text{cooling,CT}} = P_{\text{IT}} + P_{\text{CRAC}} + P_{\text{CH}}, \\
\mu_{\text{CT}} = \frac{T_{\text{CWR}} - T_{\text{CWS}}}{T_{\text{CWS}} - T_{\text{wb}}}, \\
\mu_{\text{CT}} = \mu_{\text{CT,max}}\theta_{\text{CT}}.
\end{cases}
\tag{3.11}
$$

These equations can implicitly summarized as a function P_{CS}. Note that three variables (T, φ and p) are coming from weather information, and also T_{CR} is set to $25°C$. Therefore, we have $25 - 17 - 3 - 1 = 4$ variables to be set by design or by the management mechanism and controller. We choose to fix the value of one of these variables: $T_{\text{CWS}} = 30°C$ which is the temperature of cold water coming from CT to chiller units. Three control variables remain that will be determined by the optimization process: θ_{CRAC}, θ_{CH}, and θ_{CT}. Therefore, the function P_{CS} can be expressed as

$$
P_{\text{CS}} = P_{\text{CS}}\left(P_{\text{IT}}, \theta_{\text{CRAC}}, \theta_{\text{CH}}, \theta_{\text{CT}}\right) \tag{3.12}
$$

where P_{IT} is the IT equipment's consumption as defined before.

As it was mentioned before, the cooling system is locally optimized in the design of this research. At each moment, the cooling system optimizer of each data center receives the P_{IT} from the server models which provide the amount of heat produced in that data center, and also receives the weather information from the weather stations which provides the outdoor (dry bulb) temperature T, relative humidity φ and outdoor air pressure p, summarized as the T_{wb} for the system of equations. The optimization problem of the cooling system of a data center can be formulated as follow:

$$\begin{aligned} & P_{\text{cooling system}} = P_{\text{CS}}\left(P_{\text{IT}}, \theta_{\text{CRAC}}, \theta_{\text{CH}}, \theta_{\text{CT}}\right) \\ & \text{subject to} \\ & 0 \leq \theta_{\text{CRAC}}, \theta_{\text{CH}}, \theta_{\text{CT}} \leq 1 \end{aligned} \tag{3.13}$$

Any nonlinear numerical method can be used to implicitly solve the system of equation and locally optimize the resulting function in each data center. Matlab optimization toolbox is used in this research to solve this optimization problem. The result of each local optimizer is the energy consumption of the cooling system in each period of time, which will be in turn reflected by the PUE measure in the experimental results.

3.2.3 Summary

In this section, a complete model for cooling system of a data center is presented. It is also identified what parameters should be adjusted in order to minimize the energy consumption of the cooling system, and also how these adjustments should be performed.

3.3 Chapter Summary

In this chapter, three important models for energy consumption and carbon footprint of data centers are presented. These models are associated with cooling system, server, and profit-per-service. The cooling system and profit-per-service models are used in the simulation platform for HPC job scheduling, and server model is used in the simulation platform of web application load balancing.

CHAPTER 4

CARBON-PROFIT-AWARE JOB SCHEDULER

In Chapter 2, main concept of a new design for CPA Geo-DisC system was introduced. The most important module of such system is its job scheduler. The system objectives will be reflected through this module, and if this module works efficiently, then the system can achieve its goal(s) thoroughly.

In the following sections, first, scheduling metrics will be defined. Next, the scheduling algorithm will be proposed, and last, foreseeable outcomes of the scheduler will be described.

4.1 Scheduling Metrics

Before describing the CPA scheduler, it is important to recognize its metrics. In the following, the metrics which are used in the CPA scheduler are listed and explained:

- Minimum Completion Time (MCT): The absolute time, which a job is scheduled to finish. The purpose of this metric is to maximize the system's performance by scheduling the jobs, as much as possible.
- Minimum Carbon Emission (MCE): This metric is based on the greenness of the data centers, and it is used to schedule the jobs on the available servers with minimum carbon emissions.
- Energy Price: This metric measures the energy price of each data center as a decision factor.
- Deadline: The deadline of jobs.

In the following subsections, various topics related to predicting the behaviour of the system in its near future (24 hours ahead) in terms of several metrics, impact of imposing a carbon tax, and introducing a new metric designed for profit optimization purposes will be detailed.

4.1.1 Energy and Carbon

Energy consumption and carbon footprint metering of the entire system, servers, and jobs are very important topics. If the definitions of power and energy[1] are not accurate, it can lead the system to the wrong directions by bad decisions. In addition to the energy consumption and carbon footprint of the IT equipments, the support system has a significant contribution to the energy consumption and carbon footprint of the whole system such as a cooling system, which is one of the biggest elements of an IT infrastructure's supply system. To lower the energy consumption and carbon footprint of the cooling system, they must be accurately modeled and optimized.

4.1.2 Carbon Tax

In some states such as Australia and California, there are already some regulations in place by governments in order to control and reduce the carbon emissions. According to the system models, it is possible to calculate the carbon footprint of the entire system, but this total carbon footprint is the result of several data centers contributions which are located in different states with different carbon regulations, presumably. Therefore, the share of carbon footprint associated to each region of NDC with different carbon regulations need to be calculated separately, and then the related carbon regulations be applied to each part. In the experimental result chapter, a flat carbon tax rate is considered for the data centers located in regions with carbon regulations.

4.1.3 Profit per Core-Hour-GHz

When all above discussed metrics are available, it is possible to calculate a new metric, Profit per Core-Hour-GHz (PpCHG), which was defined in the modeling section (Section 3.1.1). As it was stated before, PpCHG(f) shows how much profit is obtained by deducting the operational cost from the amount of sale associated with running a core of CPU with adjusted frequency (f) for an hour.

[1] refer to Appendix I.3 for details.

This metric can be determined for any time of the day and any of the servers in the cloud. If real values of parameters are not present and predicted values are used for calculation of PpCHG, its value will also be a prediction for that particular time and core of the server, which can be used in the CPA scheduler algorithm. Nevertheless, the outcomes of the scheduler may degrade based on the error margin of the predictions.

Unlike MCT, MCE and other metrics used in job scheduling algorithms, in a geographically distributed cloud, PpCHG directly calculate the profit per core-hour of a server which can be optimized when the profit is one of the objectives of the system. In fact, the PpCHG draw a guiding map for the scheduler for all the cores and times of the system and not for jobs as it is a common practice in other algorithms. In Figure 4.1, a sample map is illustrated which shows the maximum profit achievable in each core at any time within a time window. As it is shown in the figure, some areas are darker then other areas which indicate a lower possible profit in those areas that should be avoided by the scheduler or, if avoiding is not possible, by reducing the frequency of the CPUs in those areas. It is also shown that there are some dark areas following by brighter areas and following by dark areas. Those dark areas represent the time of day that the energy price is high and therefore the profit of the system is low in those regions. Some cores are generally darker than other cores which indicate that the region that they are placed in either has a high energy price or the carbon tax is high and the energy mix is not so green.

The map uses a color code to indicate the relevance of optimum frequency and its associated profit. The color code is defined in Figure 4.2, where both frequency and profit are normalized, blue indicates maximum profit with minimum frequency, red indicates minimum profit with maximum frequency, magenta indicates maximum profit with maximum frequency, and no-color indicates minimum profit with minimum frequency. All other intermediate colors show a linear relation with maximum profit and frequency. The black dots on the color map shows a typical CPA scheduler actual states which are around the green line which is the optimal value of frequency. More details are provided in Section 4.3.1.

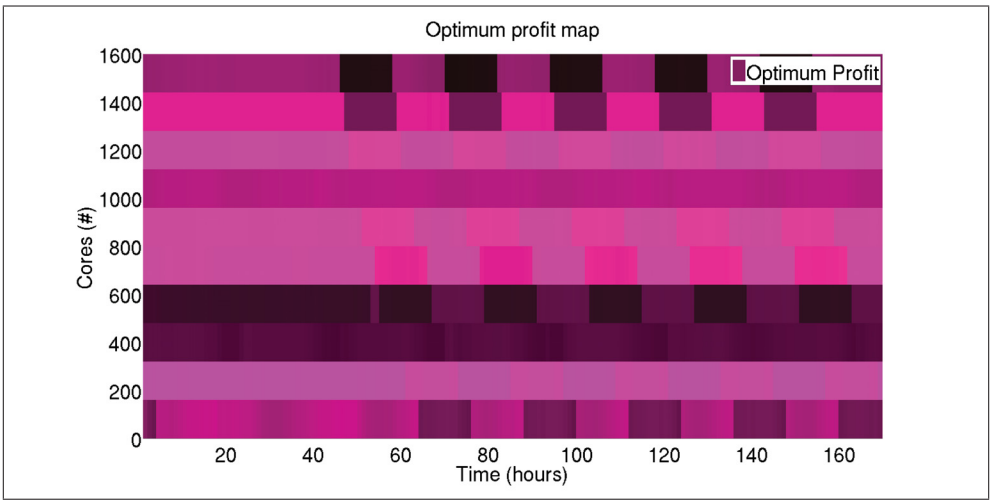


Figure 4.1 Geo-DisC maximum profit per core-hour

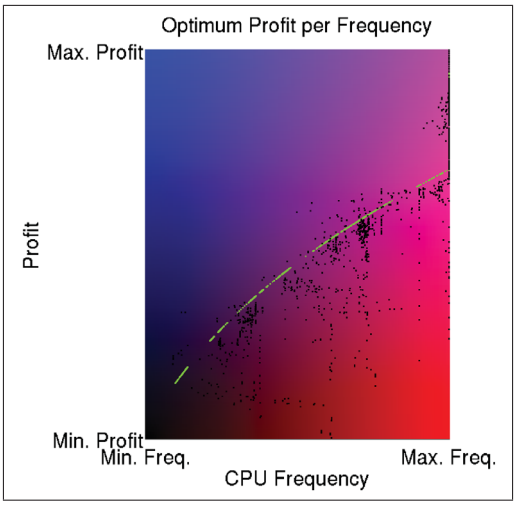


Figure 4.2 Profit per core-hour color code

4.1.4 Summary

In this section, the main metrics of Geo-DisC schedulers were described which includes MCT, MCE, energy price, jobs deadline, energy consumption, carbon footprint, cooling system energy consumption, carbon tax, and profit per core-hour-ghz. In a scheduler, some or all of these metrics might be used to decide the position of the job trace. Each metric might have a significant impact on the result of the system which these impacts are explored in the section 6.

In the next section, the basis of the CPA scheduler is described which will use all the metrics defined in this section as part of its decision factors.

4.2 Optimization Problem

With having the PpCHG metric which is the profit of the system for running a core of CPU for an hour, it is possible to calculate the whole profit of the system by making summation on all the cores and hours. In the following, the optimization problem is defined based on maximizing this summation.

$$\max_{X} \sum_{c \in C} \sum_{h \in H} \text{PpCHG}_{x_{c,h}} \tag{4.1}$$

subject to

$$\begin{aligned}
&h(y_{j,h'_j,c'_j}) = h(y_{j,h'_j+1,c'_j}) + 1, \forall h'_j = 1, ..., {h'_j}^{\max} - 1; c'_j = 1, ..., {c'_j}^{\max} \\
&c(y_{j,h'_j,c'_j}) = c(y_{j,h'_j+1,c'_j}), \forall h'_j = 1, ..., {h'_j}^{\max} - 1; c'_j = 1, ..., {c'_j}^{\max} \\
&s(c(y_{j,h'_j,c'_j})) = s(c(y_{j,h"_j,c"_j})), \forall h'_j, h"_j = 1, ..., {h'_j}^{\max} - 1; c'_j, c"_j = 1, ..., {c'_j}^{\max} \\
&h(y_{j,{h'_j}^{\max},.}) \leq e_j; h(y_{j,1,.}) \geq b_j, \forall j = 1, ..., j_{\max}
\end{aligned} \tag{4.2}$$

where $X = \{x_{c,h} | c \in C, h \in H\}$, $x_{c,h} \in Y \times F$ $(\mathbb{Z}^{*3}\mathbb{N})$, $C = \{1, ..., C_{\max}\}$, $H = \{1, ..., H_{\max}\}$, $F = \{f_{\min}, ..., f_{\max}\}$, $Y = \{(j, h'_j, c'_j) | j = 1, ..., j_{\max}; h'_j = 1, ..., {h'_j}^{\max}; c'_j = 1, ..., {c'_j}^{\max}\}$, and $h(y)$ returns the associated hour of the y, $c(y)$ returns the associated core of the y, and $s(y)$ returns the associated server of the y. The last constraint is limiting the jobs to be scheduled before their entry time (b_j) and after their deadline (e_j). The rest of the constraints are for keeping the pieces of a job (y_{j,h'_j,c'_j}) together. Worth noting that X represent the whole core-hour space of the system and Y represents the same space for each job. The scheduler job is to assign the Y and F to X while maximizing the summation of the profit.

4.3 CPA Scheduler Algorithm

In the previous section, some metrics of HPC job scheduling were presented. In this section, an algorithm is introduced in order to use those metrics for the optimization of the Geo-DisC system. The main design behind this Carbon-Profit-Aware scheduler is to optimize the frequency of the running CPUs to a value which the profit is maximized. As discussed in the literature review section, some existing algorithms calculate an optimum frequency with the lowest energy consumption, but as it will be shown in the experimental result section, optimizing the frequency to minimize the energy consumption of jobs separately will not necessary lead the scheduler towards the best profit and lowest carbon footprint. This is simply because of the presence of a large number of parameters and metrics in the decision process which makes the system very complex, and greedy actions like optimizing the energy consumption of jobs individually cannot ensure a global optimization.

Considering the requested job trace, it is consisted of a number of jobs with different number of cores, various default lengths, different entry times, and various deadlines. If a server is considered, for each distinct time in the future, it has a certain amount of energy consumption, various energy mixes, various energy prices, and a certain amount of support system energy consumption which may or may not be known to the scheduler at the moment. There are also some parameters which are most likely fixed during the operation of the scheduler such as carbon tax, sales tax, and corporation tax. The CPA scheduler need to have access to all these information in order to work properly. If the value of any of these parameters is not known to the scheduler, it will use its own predictors to estimate that value.

As mentioned above, the main idea behind proposed scheduler is to optimize the PpCHG where the variable is the frequency of the CPU. This variable can vary between minimum possible frequency and maximum possible frequency of that CPU model. This calculated optimum frequency for each cell of core-hour will be used to schedule the jobs. As it is impossible to estimate this frequency for all times in the future, a time window is selected (for example 24 hours) for the scheduler in which the scheduler will calculate the best frequency of the servers cores. The related calculation of optimum frequency is provided in the next section. As the op-

timization is to maximize the profit, it might seem that the scheduler only assure the maximum profit, but not the minimum carbon footprint. In fact, the carbon footprint minimization objective is considered in the profit per Core-Hour-GHz metric by the introduction of the carbon tax. When the carbon tax is higher, then the profit will decrease, when the server greenness is low. Therefore, the frequency of the server will be adjusted in a way which guarantee the maximum profit and minimum carbon footprint. As not all the states in the world practice the carbon footprint regulations, forcing the scheduler to minimize the carbon footprint in these states might be difficult. Consequently, in this research another parameter will be introduced as Virtual Carbon Tax (VCT) which will virtually force the scheduler to minimize the carbon footprint without affecting the normal operation of the system. Full details of the virtual carbon tax and its impact on the CPA scheduler are provided in the following sections.

The following pseudo code describe the main module of the scheduler in an abstract level.

```
1: repeat
2:   calculate optimum PpCHG and f_opt for all free slots.
3:   for job_i do
4:     find all compatible_free_slots with job_i.
5:     sort (job_i,compatible_free_slot) pairs.
6:     select the best pair based on PpCHG,MCE,MCT(f_opt).
7:     insert the best pair to the best_pair_list.
8:   end for
9:   sort best_pair_list.
10:  select the best pair based on PpCHG,MCE,MCT(f_opt).
11:  schedule the best best_pair.
12:  remove the scheduled job from the job_list.
13:  update the free_slots.
14: until (no more job) OR (no more suitable free slots)
```

As it is described in this pseudo code, first, the list of compatible jobs and free slots will be extracted. Then, the optimum frequency will be calculated for each free slot. Next, pairs of each job and its compatible slots will be sorted based on MCE, MCT, and PpCHG. Worth

noting that the MCT metric changes based on the value of the optimum frequency. Then, the best pair will be added to list. This process will be done for all the jobs. Next, the list of best pairs will be sorted again based on the similar metrics. Then, the best pair of this list will be selected to be scheduled. A VM will be scheduled to be created and run the associated job with the number of requested core and the optimum frequency. The whole process will be repeated until there is no more unscheduled job left or there is no compatible slot left.

4.3.1 Optimum Frequency Calculation

As it was mentioned in the previous section, in this scheduler, a new metric is used to maximize the profit of the system, PpCHG. Here, we discuss how to use this metric to obtain the optimum working frequency for each CPU core in the system. Equation 3.2 could be rewritten in a polynomial form in respect to the f_{cpu} as Equation (4.3).

$$\begin{aligned}
\text{PpCHG} &= (1 - \text{TAXR}_{\text{corp}_{\text{region,effective}}}) * \text{PUE}_{\text{average}} * \Bigg(- \beta_{\text{cpu}}\text{EP}_{\text{region}} \\
&- (1 - g_d(t)) * \rho_{\text{max}} * \beta_{\text{cpu}}\text{CTR}_{\text{region}} \Bigg) \\
&- (1 - \text{TAXR}_{\text{corp}_{\text{region,effective}}}) * \text{OPEX}_{\text{core,hour}} \\
&+ (1 - \text{TAXR}_{\text{corp}_{\text{region,effective}}}) \Bigg(\text{SR}_{\text{core,hour,gig}} \\
&- \text{SR}_{\text{core,hour,Gig}} * \text{TAXR}_{\text{sales}_{\text{region}}} \Bigg) * f_{\text{cpu}} \\
&+ (1 - \text{TAXR}_{\text{corp}_{\text{region,effective}}}) * \text{PUE}_{\text{average}} * \Bigg(- \alpha_{\text{cpu}}\text{EP}_{\text{region}} \\
&- (1 - g_d(t)) * \rho_{\text{max}} * \alpha_{\text{cpu}} * \text{CTR}_{\text{region}} \Bigg) * f_{\text{cpu}}^3
\end{aligned} \tag{4.3}$$

Because the coefficient of f_{cpu} is a positive value and coefficient of f_{cpu}^3 is a negative value, the graph of PpCHG(f_{cpu}) have one maximum for $f_{\text{cpu}} > 0$ as it is illustrated in the Figure 4.3. To calculate this maximum point, it is sufficient to calculate its positive root from

$\text{PpCHG}'(f_{\text{cpu}}) = 0$ as it is shown in Equation (4.4), where its optimum frequency can be calculated from Equation (4.5) and (4.6).

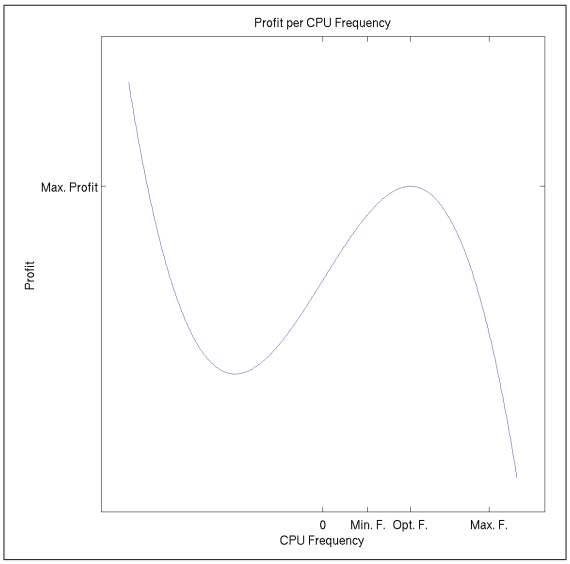


Figure 4.3 Profit per CPU frequency graph

$$
\begin{aligned}
& (1 - \text{TAXR}_{\text{corp}_{\text{region,effective}}}) \Bigg(\text{SR}_{\text{core,hour,gig}} \\
- \quad & \text{SR}_{\text{core,hour,Gig}} * \text{TAXR}_{\text{sales}_{\text{region}}} \Bigg) \\
+ \quad & 3 * (1 - \text{TAXR}_{\text{corp}_{\text{region,effective}}}) * \text{PUE}_{\text{average}} * \Bigg(- \alpha_{\text{cpu}} \text{EP}_{\text{region}} \\
- \quad & (1 - g_d(t)) * \rho_{\max} * \alpha_{\text{cpu}} * \text{CTR}_{\text{region}} \Bigg) * f_{\text{cpu}}^2 \qquad = 0
\end{aligned}
\tag{4.4}
$$

$$
f_{\text{cpu}} = \sqrt{\frac{\text{SR}_{\text{core,hour,gig}}(1 - \text{TAXR}_{\text{sales}_{\text{region}}})}{3 * \text{PUE}_{\text{average}} * \alpha_{\text{cpu}} * (\text{EP}_{\text{region}} + (1 - g_d(t)) * \rho_{\max} * \text{CTR}_{\text{region}})}} \tag{4.5}
$$

$$f_{\text{cpu}_{\text{optimum}}} = \begin{cases} f\text{min} & \text{if } f_{\text{cpu}} \leq f\text{min} \\ f_{\text{cpu}} & \text{if } f\text{min} < f_{\text{cpu}} < f\text{max} \\ f\text{max} & \text{if } f_{\text{cpu}} \geq f\text{max} \end{cases} \tag{4.6}$$

As Equation (4.5) shows, the optimum frequency of the CPU is based on several parameters of the system such as $\text{SR}_{\text{core,hour,gig}}$, $\text{TAXR}_{\text{sales}_{\text{region}}}$, $\text{PUE}_{\text{average}}$, α_{cpu}, $\text{EP}_{\text{region}}$, $g_d(t)$, and $\text{CTR}_{\text{region}}$, which shows the dependency of the optimum frequency and market price for HPC jobs, sales tax of the region, weather and temperature variations in the region, model of CPU, energy price in the region, and carbon regulations in the region, respectively. These relations will be examined in detail later in the experimental results chapter (Chapter 6). There was another parameter in the equations which did not show in the optimum frequency equation, $\text{TAXR}_{\text{corp}_{\text{region,effective}}}$. It seams that optimum frequency is not affected by the corporation tax. However, this parameter besides other parameters have an effect on the absolute value of the profit, which can be positive or negative. Therefore, this parameter could be part of a trigger for the scheduler algorithm to refuse to schedule a job in a time slot.

Here, the relation between the optimum frequency parameters and the optimum frequency will be studied in order to provide enough knowledge for better understanding of its behavior in the experimental environment. In the following Figures, the optimum value for CPU frequency is calculated based on optimization of PpCHG. Since, there are many parameters in the PpCHG, a color code is used to cover some of these parameters as it is illustrated in Figure 4.4. The color codes are presented for different energy price, CPU frequency, and server greenness.The same color code is also used to illustrate the schedule of jobs in the experimental result section.

The frequency of CPUs is illustrated by the red element of color. The CPUs with the minimum frequency have no red color and those with the maximum frequency are full red. The frequencies in between the minimum and maximum are shown with a radian of red. The blue and green elements of the color are representing the price of energy and greenness of the server. Full blue means lowest energy price and no blue means highest energy price in the region which server

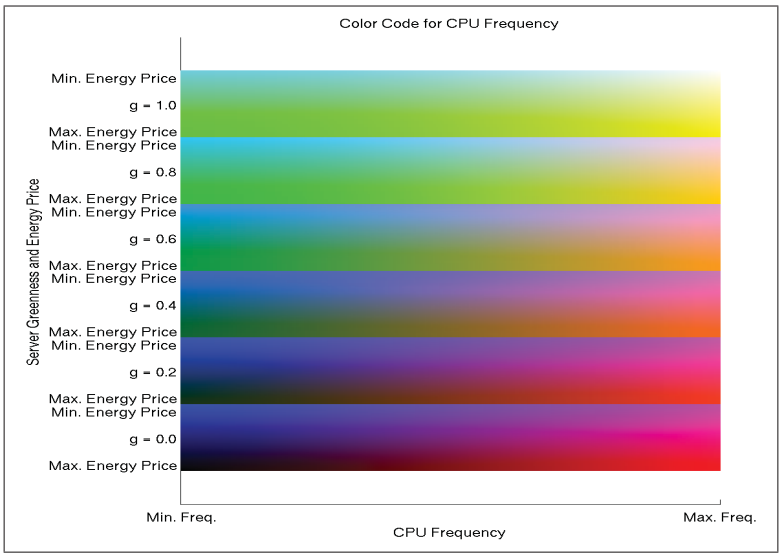


Figure 4.4 Color code of scheduled jobs

is located. No green means that the server is running on the dirtiest type of energy and full green means the server is running on completely renewable type of energy. The color codes are selected in a way that brighter colors are more near to the objective of the system and darker colors are far from it, in general. For example, a full green, red, and blue color has the ideal condition for job scheduling, because the job is running at the maximum frequency which is good for the performance of the system, and it is running on a completely green energy source which has no carbon footprint, and the energy price is at its lowest which helps to maximize the profit. With similar reasons, no-color has the worst condition for job scheduling.

As it is shown in the color code, each color in the color space has a meaning for the scheduled job. For example, a scheduled job with purple color means that the job is running on a completely non-green server, but the price of electricity is low and the job is scheduled to work on CPUs with maximum frequency. The Table 4.1 present the main features of main colors, and their impacts on scheduler metrics.

Based on this table, white is the ideal color, and no-color, red, blue, and yellow are not fit with the objectives of the system. Magenta and green are to some extent acceptable, but cyan is not acceptable because in the cyan case, the frequency of the server can be adjusted to maximum

color code	meaning	energy impact	carbon impact	profit impact
No-color	Job is running on non-green servers with minimum frequency and energy price is high	low increase	low increase	decrease
White	Job is running on green servers with maximum frequency and energy price is low	increase	no increase	increase
Red	Job is running on non-green servers with maximum frequency and energy price is high	increase	increase	decrease
Blue	Job is running on non-green servers with minimum frequency and energy price is low	low increase	low increase	low increase/decrease
Green	Job is running on green servers with minimum frequency and energy price is high	low increase	no increase	low increase/decrease
Magenta	Job is running on non-green servers with maximum frequency and energy price is low	increase	increase	low increase
Yellow	Job is running on green servers with maximum frequency and energy price is high	increase	no increase	low increase/decrease
Cyan	Job is running on green servers with minimum frequency and energy price is low	low increase	no increase	low increase

Table 4.1 A color code describing the status of the scheduled jobs

and increase the profit without worrying about the carbon footprint and energy price. To have a better understanding of the optimum points, a typical system is optimized for maximum profit and the optimum frequencies are illustrated with white color in the Figure 4.5. As it is shown, the optimum value for frequency of the CPUs is variable based on energy price and greenness of the servers. In general, in non-green servers (the lowest part of the figure), when the energy price is low, the optimum point stands on high frequency (bright magenta), and with the rise of the energy price, the optimum point stands on lower frequencies (darker magenta). The optimum point is far from both side of high frequency-high energy price and low frequency-low energy price. Same conclusion is correct for other parts of the figure with greater greenness. For example, in the section with the highest greenness (the highest part of the figure), the optimum frequency happens far from yellow and cyan colors and remain near green to white color. The only obvious difference between the lowest greenness and highest greenness part is that the optimum frequency occurs in the higher values when the system is greener.

In the Figures 4.6, 4.7, and 4.8, the optimum frequency is achieved for a typical system under variation of three parameters, energy price (EP), carbon tax (CT), and sales rate for a core-hour (SR). As it is illustrated, when the EP and CT increase the optimum frequency decrease, and when the SR increase the optimum frequency increase. As it is shown in the Figure 4.8, if the SR is high enough it will cover for the EP and CT and the optimum frequency lean to maximum frequency, but businesses cannot increase the SR arbitrarily if the competitiveness exist in the market.

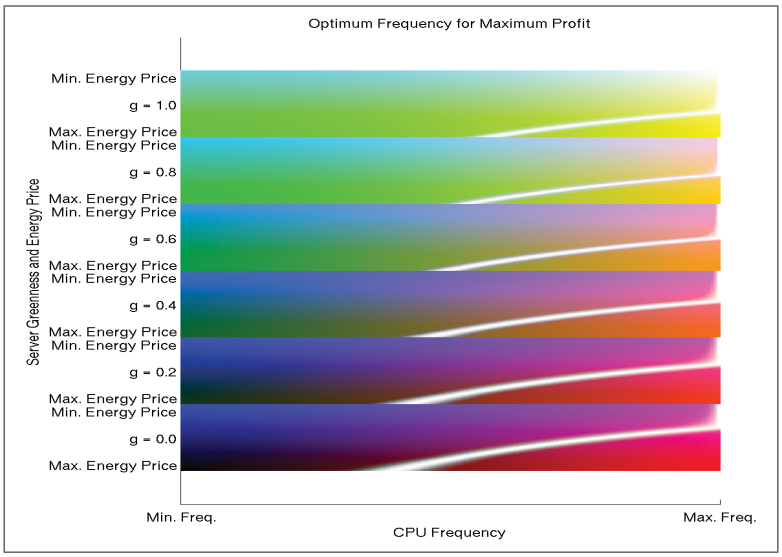


Figure 4.5 Optimum frequency for maximum profit

4.3.2 Virtual Carbon Tax

In the previous section, it was shown how to find the optimum frequency of the CPUs in order to maximize profit. In that case, carbon tax is considered as a parameter which automatically minimize the carbon footprint of the system in effect of minimizing the cost and maximizing the profit, but in many states the carbon tax is not implemented yet or is very little. Therefore, in these states the pressure of the carbon tax for minimizing the carbon footprint of the system does not exist.

Because the carbon footprint is a high profile and sensitive matter in societies, even though the carbon tax may not exist, businesses may choose to reduce their carbon footprint voluntary for a good public figure (i.e. consumers' beliefs have a direct impact on corporations share price). Therefore, in this subsection, a method is introduced to create a multi-objective system for maximizing the profit while considering the carbon footprint of the system voluntary. The main idea of this method is based on the introduction of a virtual carbon tax.

Virtual carbon tax is a particularly similar concept to the carbon tax. Based on the carbon footprint of the system, a tax rate will be applied to the amount of carbon producing by the system. From outside of the business, this money is part of the profit and is taxable, so in the

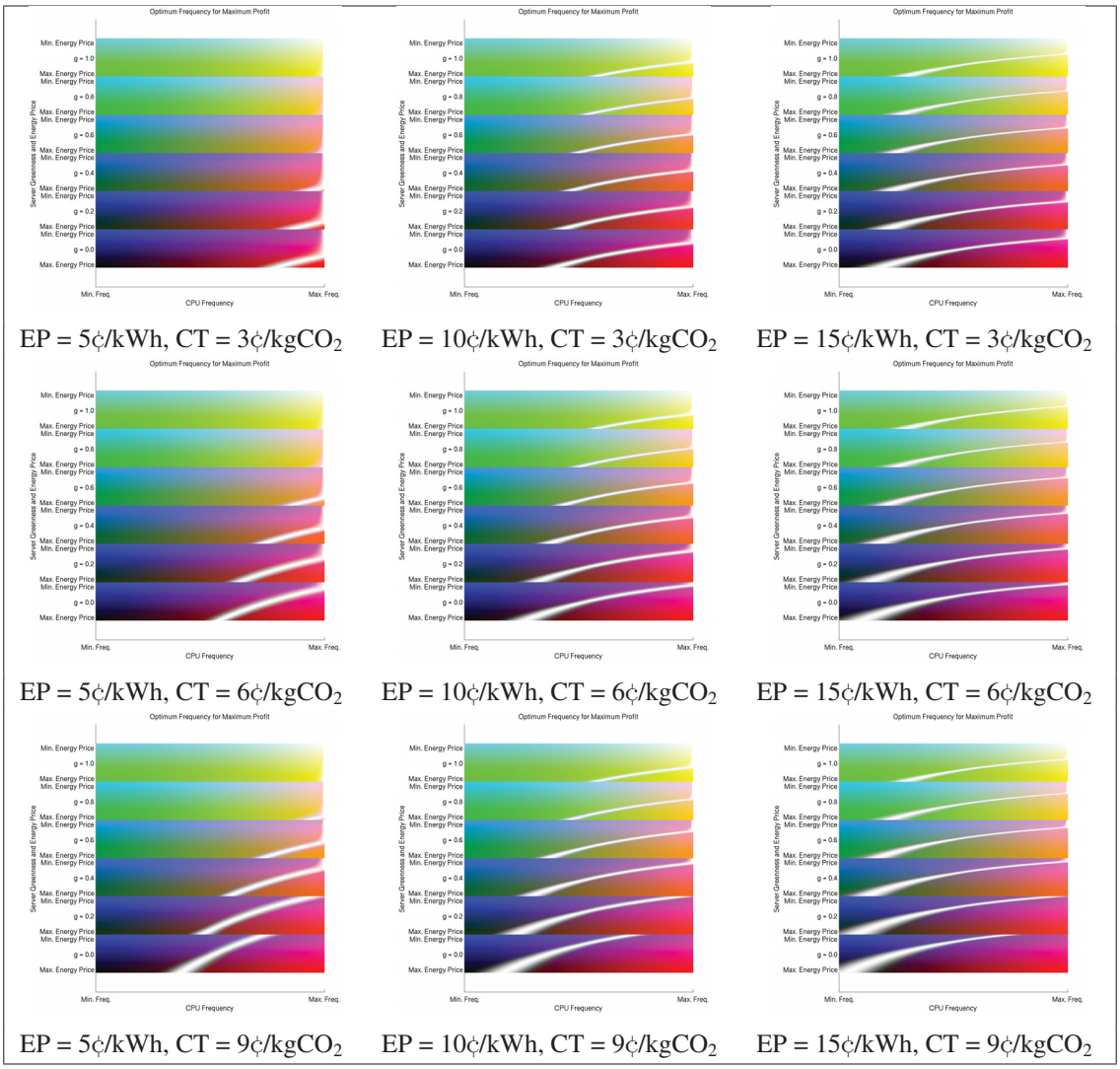


Figure 4.6 Optimum CPU frequency with sale rate = 2¢ per core-hour

calculations, corporation tax will apply to this amount. In fact, when the virtual tax serves its purpose in the scheduler, the amount of virtual carbon tax plus the profit of the system will create the real profit of the corporation. The only thing that virtual carbon tax is doing is to create a virtual need for carbon reduction in the cover of cost reduction. If the algorithm is not carbon sensitive, the introduction of virtual carbon tax has no effect. The VCT work as a catalyst[2], and has no real world existence, but it forces the carbon sensitive algorithms to

[2]In chemistry, catalyst is a substance which increase the rate of reaction, but remain unchanged at the end of the reaction. Similarly, VCT acts as a carbon tax to decrease the carbon footprint of the system, but at the end it is part of the profit.

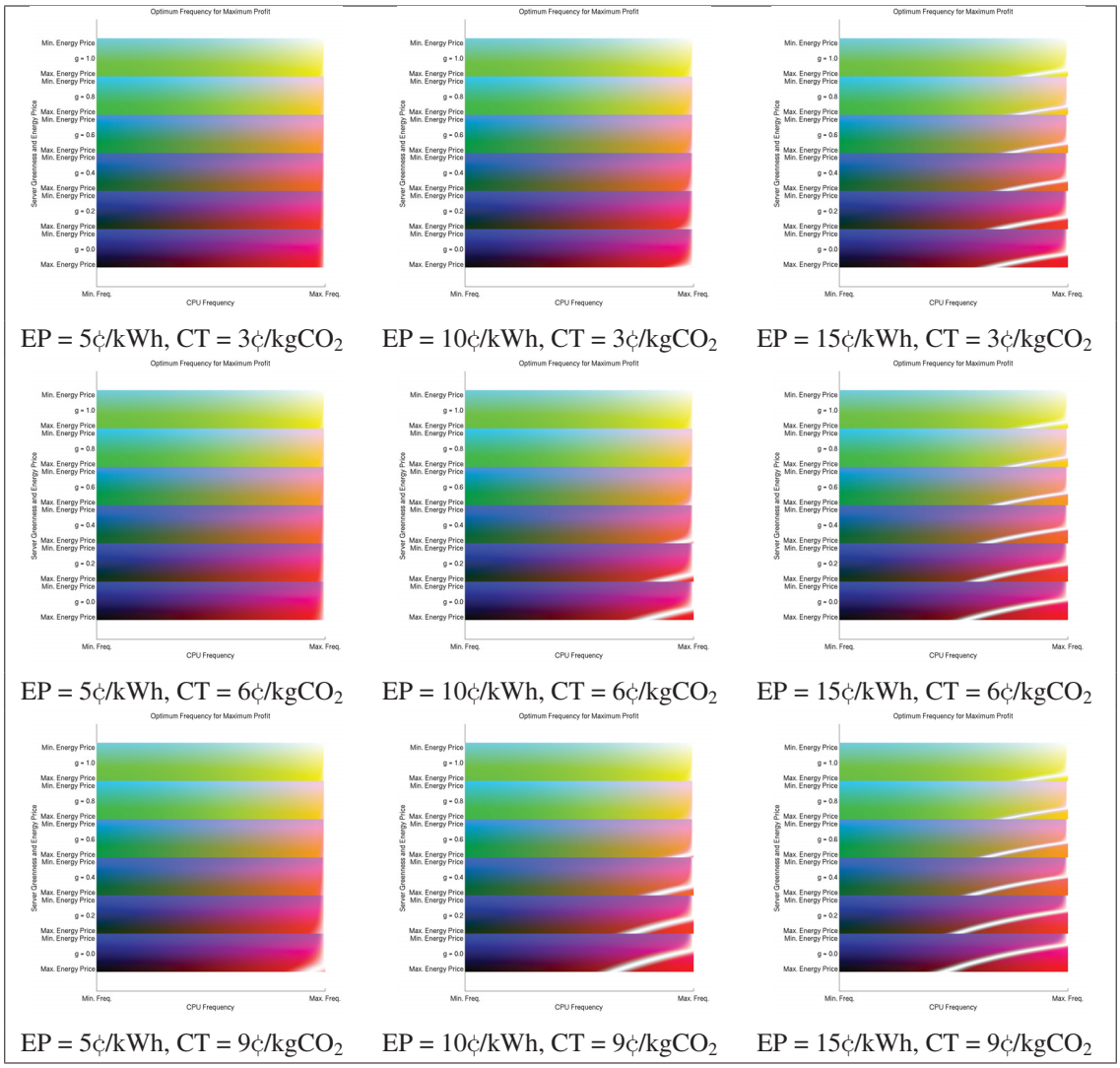


Figure 4.7 Optimum CPU frequency with sale rate = 4¢ per core-hour

consider more carbon reduction while those algorithms maximizing the profit. Last, the VCT will be combined with the net profit. Some corporations may choose to allocate part of VCT money for environment-friendly projects, which may bring them some tax breaks, as well.

In Figure 4.9-a, the cost breakdown of a typical system is presented. In the following, Figures 4.9-b and 4.9-c show that same system with the same specifications with application of virtual carbon tax. These figures are the same except for the position of VCT in the graph. Figure 4.9-b shows how introduction of VCT increases the volume of total carbon tax (CT plus VCT), which will put pressure on carbon sensitive algorithms. The effect of VCT on CPA algorithm

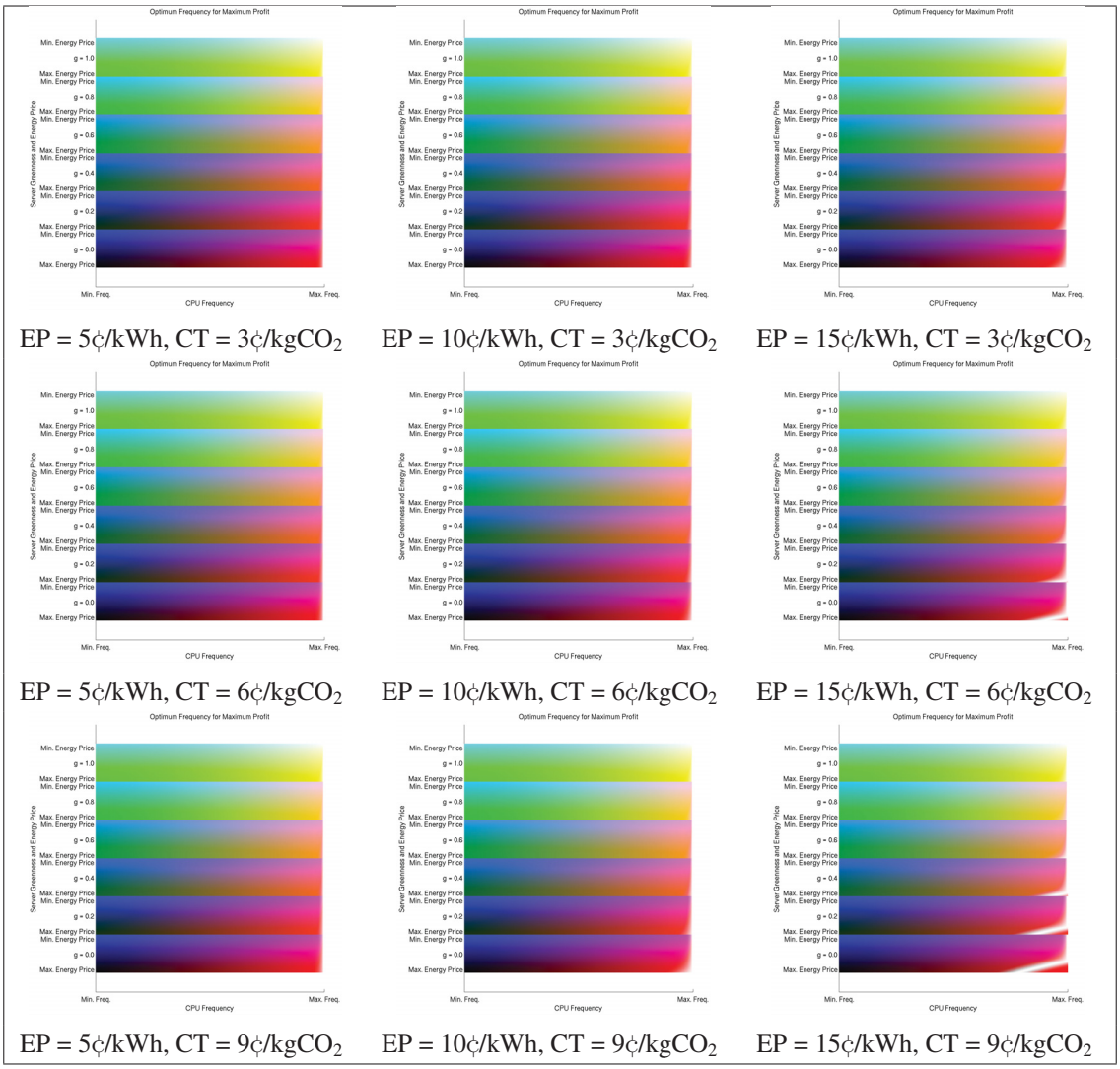


Figure 4.8 Optimum CPU frequency with sale rate = 6¢ per core-hour

is examined in the experimental result chapter. Figure 4.9-c shows how combination of VCT and system profit create the real profit of the system (need to be compared with Figure 4.9-a).

4.3.3 Summary

The basis of the CPA scheduler is introduced in this section. The CPA scheduler work based on a new metric: profit per Core-Hour-GHz. This metric is optimized ahead of the time with real and predicted values for a time window in the future. Then, the scheduler will use a greedy

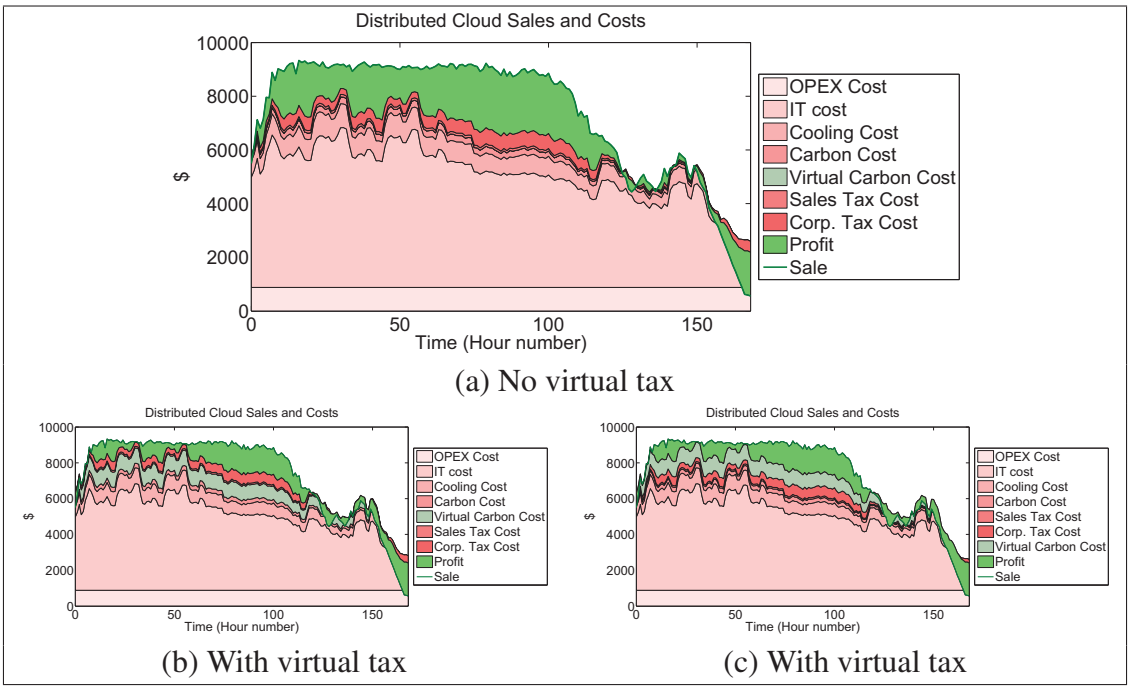


Figure 4.9 Cost breakdown of a typical system

approach to schedule the jobs which fit the best with this metric and other metrics such as MCE. Next section will describe the expected outcomes of the proposed scheduler.

4.4 Expected Outcome

Based on the description of the scheduler provided in the previous section, certain behaviours are expected to be observed by this scheduler under certain circumstances. This section will describe those events and their expected outcomes. Then expected results will be validated in the experimental results chapter.

4.4.1 Performance

Since the CPA scheduler adjust the frequency of the CPUs to achieve its goals, it is expected that the total amount of jobs done by the CPA scheduler to be less than amount of jobs done by a performance scheduler. However, when the profit or the carbon footprint of the algorithms are being compared, it is expected that the CPA scheduler have a better results.

4.4.2 Virtual Carbon Tax

Based on the definition of the Virtual Carbon Tax, it is a cost which is calculated based on the carbon footprint of the system, and aggregated at the end with the profit of the system. If the VCT is used in a non-carbon sensitive algorithm, then there will be no effect on the results of the algorithm. However, if the algorithm is carbon-cost sensitive, then the VCT forces the system towards less carbon footprint direction. Because the VCT cost will be added to the profit of the system at the end of the process, it is expected that the VCT does not affect the profit of the system in a similar way as the carbon footprint of the system.

4.5 Chapter Summary

In this chapter, the main characteristic of CPA scheduler is described. From a concept point of view, there are some concepts related to data collection and some to decision making. For the data collection, common and new metrics used in the CPA scheduler are described, and then they are used in the CPA scheduler which decides for the position of the jobs in the job scheduling part of the system.

CHAPTER 5

CARBON-AWARE LOAD BALANCER

In introduction, it was mentioned that one goal of this thesis is to introduce a mechanism for load balancing of web applications. The web applications are hosted on virtual machines and those virtual machines are able to seamlessly migrate between servers and data centers. Before introduction of the mentioned mechanism, first, in the Section 5.1, we introduce a new heuristic algorithm which will be used for load balancing of the web applications. Next, in Section 5.2, we introduce the load balancing mechanism for web applications which are aware of environmental impacts of data centers.

5.1 Multi-Level Grouping Genetic Algorithm

In the GGA, a new crossover and mutation operators were introduced in order to save the group relations between individual genes. In a similar way, here, the MLGGA crossover and MLGGA mutation operators are introduced in order to preserve the relations between groups. These operators substitute the normal GA crossover and mutation operators and work along with the other GA operators as shown in lines 6 and 7 of the following MLGGA pseudocode:

```
1: Choose initial population.
2: Evaluate each individual's fitness.
3: repeat
4:   Select individuals to reproduce.
5:   Mate pairs at random.
6:   Apply MLGGA crossover operator.
7:   Apply MLGGA mutation operator.
8:   Evaluate each individual's fitness.
9: until terminating condition
```

5.1.1 MLGGA Crossover

In the virtual cloud problems, the positions of VMs are the variables of the problem. In these problems, grouped variables, such as server consolidation, lower the cost function. However, normal GA crossover break the existing groups in the parents chromosomes, and probability of preserving the good grouping features presented in parent genes is very low. Although the GGA crossover provides a way to preserve the grouping features in parent genes, there are relations between groups that the GGA crossover is not able to preserve, and most probably it breaks these relations. In the network of data centers, the GGA is good to consolidate VMs on servers, but it is not able to identify that there are benefits in choosing servers from only one data center. For example, the GGA may consolidate VMs on different servers which allow us to turn off some of the servers and save energy, but it is not aware that if it consolidate all servers on less number of data centers as well, it may save a lot more by turning off an intermittent data center. For example, assuming parent genes P1 and P2 and their groups are as follow:

P1 : ACDEGIJB
(ACDEGAIJDCBACDEAGIA)
P2 : bcghieda
(bcghieddacccceehigha)

If each group is assigned to a higher level group (a bigger bin) as follows:

$$W = \{A\}, X = \{B, C\}, Y = \{D, E, F\},$$
$$Z = \{G, H, I, J\}$$
$$w = \{a\}, x = \{b, c\}, y = \{d, e, f\}, z = \{g, h, i, j\}$$

The genes group lineup can be rewritten as their higher level groups as follows by replacing the group representations (for example, ACDEGIJB for P1) by their higher level group labels:

P1 : WXYYZZZX ← ACDEGIJB

P2 : xxzzzyyw ← bcghieda

As it is shown above, some higher level groups are repeated in the group lineup. Here, we create a higher level group lineup (level 2 group lineup), and we keep only one gene per higher level group similar to what we did in group lineup in lower level. Now, the chromosome could be written as below:

P1 : WXYZ WXYYZZZX
(ACDEGAIJDCBACDEAGIA)

P2 : xzyw xxzzzyyw
(bcghieddacccccehigha)

where the first column is the new level 2 group lineup representation of the chromosomes. The crossover will be done on the level 2 group lineup representation of the genes: (WXYZ) and (xzyw). Like the GGA, two crossover point will be chosen randomly on each gene:

P1 : WX|Y|Z WXYYZZZX
(ACDEGAIJDCBACDEAGIA)

P2 : x|zy|w xxzzzyyw
(bcghieddacccccehigha)

and the middle part of first gene will be replaced with middle part of the second gene, and replaced higher groups in first gene with their assigned groups and containing individuals will be removed from the gene. In addition, for inserted higher groups from second parent, their matching higher groups in first gene will be removed in a same way.

Offspring : WX|zy|Z

As it is shown in above, higher level group (Y) in the first parent is replaced with higher level groups (z) and (y) from second parent. This means that their matching higher level groups (Z) and (Y) are not any more valid and their containing groups (D,E,F,G,H,I,J) and their containing individuals should be removed from the chromosome; which remains the offspring chromosome as below:

Offspring : WXzy (ACghiedd?CBACehighA)

Genes number 3-8, and 14-18 in second parent (P2) are belongs to groups (d,e,f,g,h,i,j) which are belongs to higher groups (y,z) and they are transferred directly from second chromosome to the first chromosome. Gene number 9 is in group (D) in first parent which belongs to higher level group (Y) which needs to be removed as mentioned above.

For the genes in first parent, which are replaced with genes from second parents, there are some individuals which are belongs to some groups and higher level groups which are not yet removed from the chromosome. For our example, genes number 6 and 16 are belong to group (A) in first parent chromosome which are replaced with (e) and (i) from the second parent chromosome. These individuals with their co-group and co-higher-group individuals need to be removed from the chromosome as well. Co-group individuals of an individual are those genes which are in the same group, and co-higher-group individuals of and individual are those genes which are in the same higher group. For our example, all individuals in higher level group (W) which is higher level group of (A) need to be removed from the offspring chromosome. The offspring chromosome will be like this:

Offspring : Xzy (?Cghiedd?CB?Cehigh?)

As it is shown in above, higher level groups (X) from first parent, and (z) and (y) from the second parent are preserved in the offspring chromosome intact which is the goal of the crossover operator.

At the end, there are some individuals which are not assigned to any group and higher level group. These individuals will fit in the chromosome by using the First Fit Descending algorithm or more advanced fitting techniques. Higher level groups which are fuller will chosen first, and also fuller groups are in more priority for first fit algorithm.

A graphical representation of GGA was presented in previous sections. For the same parent chromosomes, MLGGA crossover and its graphical representation is illustrated in Figure 5.1, 5.2, and 5.3. There is a new color code for cross signs, which indicate the individuals belong to groups with at least one removed co-group. These individuals are identified with pink cross signs.

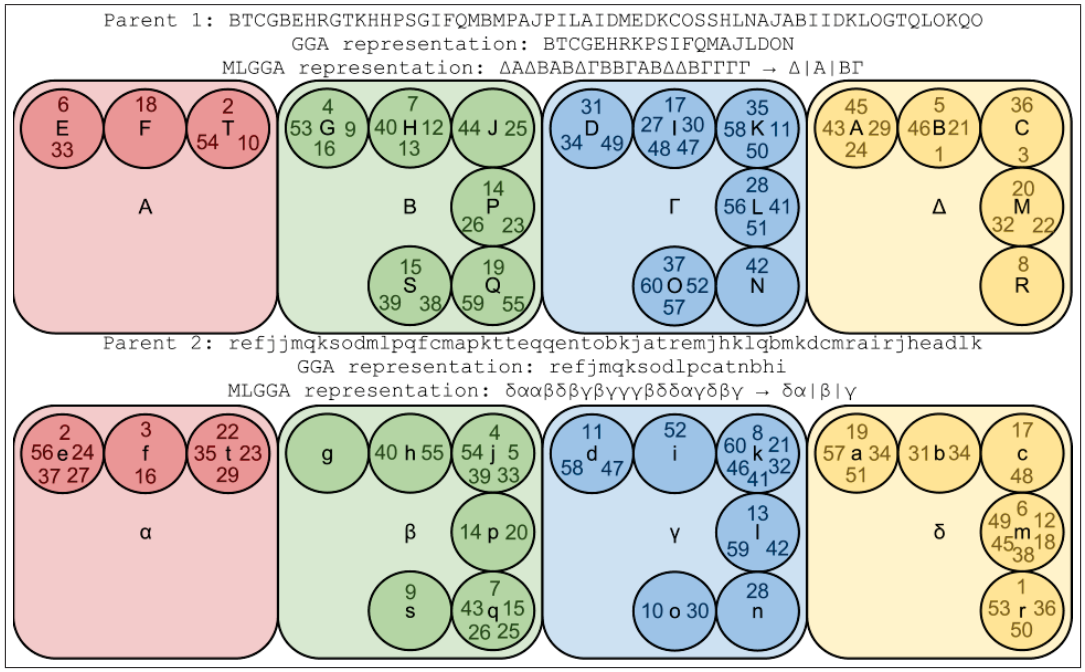


Figure 5.1 MLGGA representation for parent chromosomes.

5.1.2 MLGGA Mutation

The MLGGA mutation is very similar to the MLGGA crossover concept. From a selected chromosome:

P1 : WXYZ WXYYZZZX

(ACDEGAIJDCBACDEAGIA)

Some higher level groups will be randomly chosen, and all co-group and co-higher-group individual genes will be removed from the chromosome. For our example, if higher level group (Z) is selected to be removed, the remaining chromosome will be as below:

P1 : WXY WXYY???X

(ACDE?A??DCBACDEA??A)

Then, the First Fit algorithm will be used to reinsert them to the chromosome as described in crossover operator section.

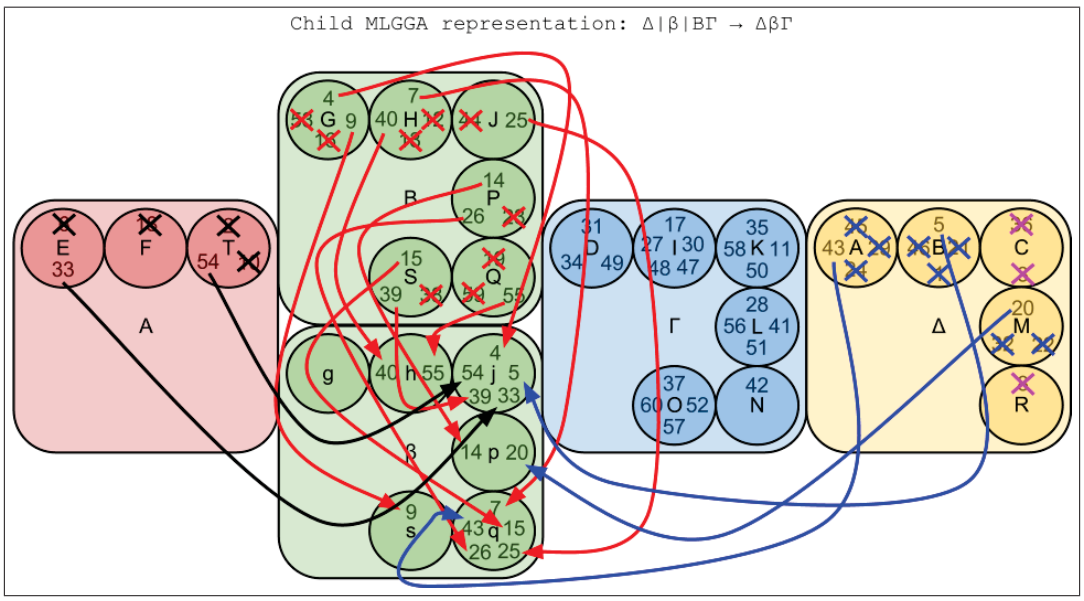


Figure 5.2 MLGGA crossover in progress.

5.1.3 Extensions of the MLGGA Crossover and Mutation

In the GGA, the concept of group of individual genes is introduced. In previous section, we described a situation where there are some relations between groups of groups in a problem.

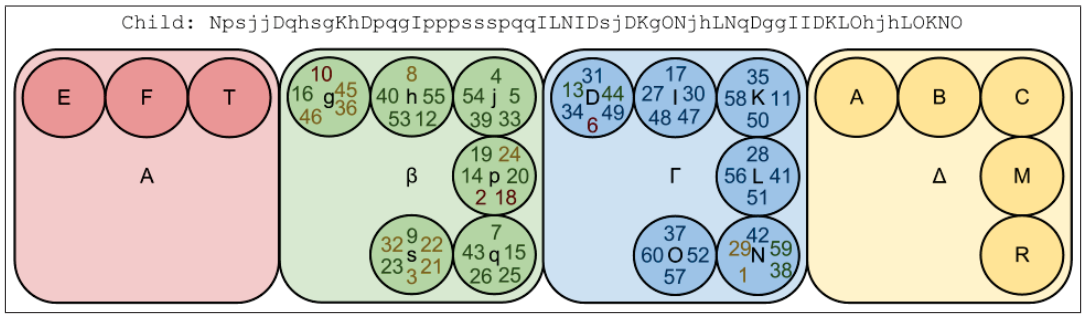


Figure 5.3 MLGGA crossover final result.

We can extend this solution for cases in which there are several level of grouping involved. For example, if, in a problem, individuals are grouped by some criteria, the problem has grouping relations at level 1. If the groups of level 1 are grouped by some other criteria, there will then be a grouping of level 2. And similarly, we can have grouping of level n for a problem.

For a problem with the grouping of level n, a level n MLGGA crossover and mutation should be used. The concept of the level-n MLGGA crossover and mutation is similar to what we described in previous subsections which was a level-2 MLGGA crossover and mutation. For the level-n MLGGA crossover, individual genes will be represent by their level 1, level 2, ..., level n groups. Two crossover point will be selected randomly in parents level-n groups representation, and the middle part of second chromosome will be inserted to the first chromosome to occupy the place of middle part of first chromosome. Their matching level-n groups in first chromosome with all their individuals will be removed from the offspring chromosome. For those individual genes in first parent which are replaced with transferring genes from second parent, all their co-level-n-group individual genes will be removed as well. Co-level-n-group individuals of an individual are those genes which are in the same level n group. At the end, all removed individuals will be inserted to the chromosome with using an First Fit algorithm or more advanced algorithms as described in previous subsections. According to this definition, the GGA algorithm is a level-1 MLGGA.

Level-n mutation operator will be defined in a very similar way with randomly selecting some level-n groups and removing their individuals and reinserting them.

5.2 Carbon-Aware Load Balancing Concept

A solution to the weaknesses of clean energy sources, such as intermittency and lack of availability, is "to follow the available clean energy" using a distributed ICT infrastructure. In this section, the components of such an infrastructure, which we call a Carbon-Aware Distributed Cloud, are discussed.

A CADCloud is a distributed cloud, in which the VM locations on its geographically-distributed servers are optimized based on the carbon footprint of the entire CADCloud. Like other distributed clouds, a CADCloud consists of a set of reliably connected data centers, which may be powered by different sources of energy, and forms an uniform environment in which seamless VM migration can be easily achieved. These sources constitute a combination of renewable and non renewable energy. Specifically, a cleanness ratio is assigned to each type of energy source. However, it is worth noting that this ratio is highly variable, even among renewable sources of energy (as well as non clean sources). Another aspect of the CADCloud is its dynamic nature. The fact that most of the renewable energy sources are intermittent makes these distributed clouds highly dynamic systems from the energy and carbon point of view.

As shown in Figure 5.4, which depicts the schema of a CADCloud, the data collector component asynchronously collects energy production, energy consumption, and resource usage statistics of each data center. This component uses different energy and carbon footprint models to create the carbon footprint cost function of the optimizer component. The controller component uses the optimizer component output, which is a new location suggested for each VM, and will instruct the distributed cloud manager to relocate VMs to their new, optimized positions, if possible.

The GreenStar Network (GSN[1]), which is an existing distributed cloud structure, is a real example of the CADCloud Concept. In the GSN, various sites are connected with high speed lightpath connections. Servers at those sites form a uniform cloud entity on the top of the

[1] http://greenstarnetwork.com

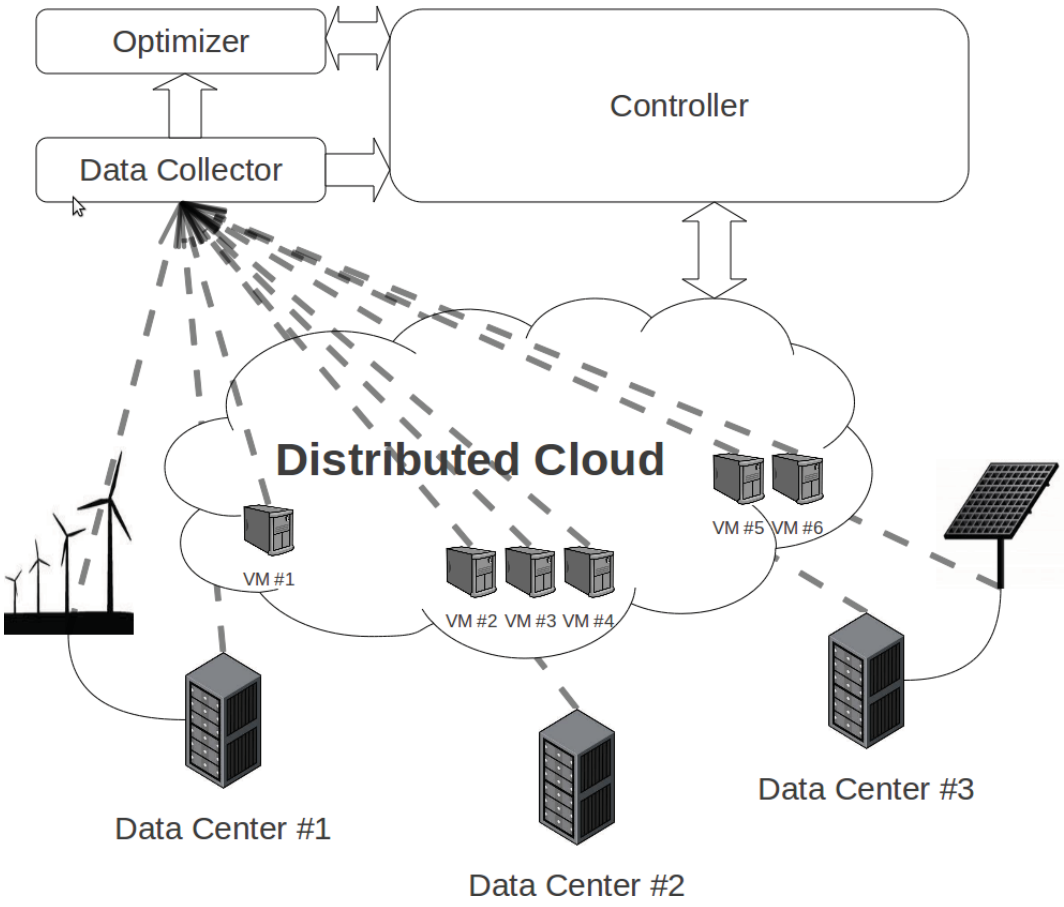


Figure 5.4 CADCloud Schema.

infrastructure. Nodes are connected to various power sources, such as hydro, solar, and wind sources.

In CADCloud, the optimizer is responsible for finding new locations for the VMs and reporting them to the controller, and it performs this task simply by using a heuristic algorithm that minimizes the cost function of the cloud. This cost function is based on the carbon footprint and energy model of individual resources in the cloud. Heuristic methods are widely used in energy efficiency solutions in cloud computing environments, and some of them are discussed in section 1.2. Since many of the algorithms currently used in the literature are mostly developed for local clouds and are not generally suitable for distributed clouds, a new algorithm is used in this work for this purpose.

An improved version of the GGA is used for a more complicated type of server consolidation in a distributed cloud that uses a diversity of energy sources (Farrahi Moghaddam *et al.*, 2012a). In that research, we introduce the Multi-Level Grouping Genetic Algorithm (MLGGA) to optimize the carbon footprint of a distributed cloud, which is formulated as a multilevel bin packing problem. In this problem, cloud servers are distributed among several data centers powered by different renewable sources of energy. Each distributed data center represents a higher level bin which includes servers. Each server represents a bin that includes VMs.

The GGA is a good candidate for one level bin packing problems (Xu and Fortes, 2010); however, for higher level bin packing problems, it is not efficient enough (Farrahi Moghaddam *et al.*, 2012a). This is because simple bin packing cannot discern higher level relations between the carbon footprint and type of energy source in data center VMs, and may not offer good solutions.

As mentioned above, there are several levels of bins in the MLGGA, and each lower level bin (server) is located in a higher level bin (data center). The MLGGA is defined by its crossover and mutation operators. The constraints for the optimization problem are adopted from (Farrahi Moghaddam *et al.*, 2011, 2012a) where server consolidation is seen as a bin packing problem.

The MLGGA performance was only compared with GGA in Farrahi Moghaddam *et al.* (2012a). In this work, it will be compared with other approaches using improved carbon and energy models. In the following sections, the MLGGA approach is used for an energy diversity study of distributed clouds.

The controller module defines the mission of the geographically-distributed cloud manager. In the GSN, the mission was to follow the sun and follow the wind, using the green energy that is available. In the CADCloud, the mission is to reduce the carbon footprint of the distributed cloud. To achieve this mission, the controller simply asks the optimizer to find new locations for VMs which yield the smallest carbon footprint. Then, the controller instructs the distributed cloud manager to move the VMs to their new locations. It is critical that the controller be able to command the hypervisors used in the distributed cloud.

5.3 Chapter Summary

In this chapter, a new genetic algorithm suitable for multi-level consolidation problems was introduced. Then, a system was introduced which uses this new algorithm and move around virtual machines to reduce the carbon footprint of the system.

CHAPTER 6

EXPERIMENTAL RESULTS AND VALIDATION

In the literature review chapter, the state-of-the-are methodologies are discussed, and the area of coverage of each research plus its pros and cons are explained. To improve those methodologies, a series of improvements are suggested in the Chapters 2, 3, 4, and 5, which they need to be evaluated and validated as a whole system.

In this chapter, the proposed methodologies in the previous chapters will be implemented in a simulation environment and compared with the baseline system, which is defined in Section 2.1. The goal of this chapter is to establish a simulation environment suitable for the context of this thesis (Obj#5) and determine the performance of the new methodologies in comparison with the state-of-the-art methodologies.

In the following sections, first, the simulation environment will be explained, and then different case studies will be investigated under the simulation environment in the subsequent sections.

6.1 Simulation Environment

This section will explain the details of the simulation environment which is used for evaluation and comparison of different designs and methods presented in the previous chapters. The scenario generator will manually or automatically alter different parameters of the system and create the simulator parameters set. Then, the simulator will execute each individual simulator parameter set separately. Each component in the system has its own running variables plus its initial parameters. The models described in the previous chapter will be used to calculate the running variables of the system in each time interval.

6.1.1 Batch Simulation

One of the issues in this research is the high number of simulations. Configuring the simulator manually for each simulation is very time consuming. Therefore, an automated batch

simulation strategy is needed to speedup the process of simulations. In the following, the main features of such strategy is described.

A simulation-job-generator is developed in this research, which is configurable to produce multiple simulation jobs at a time. For example, the simulation-job-generator will produce several jobs for a scenario which need to be simulated under different optimization algorithms. Each job is configured to run with one of the optimization algorithms.

When jobs are created, they are in a queue and will be scheduled to be executed on a sever with multiple cores by another module, simulation-job-scheduler. Simulation-job-scheduler will select simulation-jobs from the queue and will assign them to actual simulator modules. At the end, the results of simulation-jobs will be processed for production of figures and tables by two other modules, plot-generator and table-generator, respectively.

6.1.2 Caching

Models' heavy calculations are issues which slowdown the process of simulations. Based on the parameters of the system, for a model to be calculated, several intermediate parameters need to be calculated, which are very time consuming. For example, for the cooling models introduced in section 3.2, a system of 17 equations need to be solved in each iteration which may take a long time. Therefore, pre-calculated models can speedup the process of simulations.

In the development of the simulation, a dynamic approach is considered for pre-calculation of models and intermediate parameters. In this approach, the simulator will start with an empty cache, and slowly will fill it with calculated models and parameters as it goes forward. The simulator will search for pre-calculated values in the cache with a reasonable distance in the search space. If the near-enough point exist, it will use the value instantly, and if it does not exist, it will calculate the value of model for that point and will save it in the cache for future use.

6.1.3 Summary

Here, the basic concept of the test environment is explained. In the following sections, results conducted in several scenarios of this thesis will be discussed to see if they achieved their objectives. More specific simulation details related to each experiment is provided in the following sections.

6.2 Green HPC Job Scheduling Scenarios

In this section, the new Carbon-Profit-Aware HPC job scheduler which is part of the new design is tested under different scenarios. In the following sections, a number of studies are carried out on the new algorithm for the job scheduler in order to test its robustness and performance. These situations include performance, cooling system, and carbon tax studies.

6.2.1 Experimental Setup

There are various sources of the HPC-related workload traces publicly available on the Internet. A brief list of some of these sources are provided in Table 6.1. Also, some of the most cited traces are listed in Table 6.2. In this study, we use the Grid'5000 trace as a guide for our HPC trace in the experiments. This trace includes almost one million jobs. We have chosen one week of this trace to generate our HPC trace in this simulation platform, which is illustrated in Figures 6.1-a and 6.1-b, which represent the total amount of cores requested by new jobs and their average default length, respectively. As it is indicated in the figures, the HPC traces used for all of the algorithms are identical. The workload is randomly generated based on features extracted from real HPC workloads to cover fully-utilized and under-utilized situations. The workload is generated in a way that the system is fully-utilized for the first three working days of the week in the one week test scenarios and is under-utilized for the last two working days of the week.

Ten cities are selected to host the data centers in this simulation scenarios. These cities are Los Angeles, Toronto, Sao Paulo, Humburg, Cape town, Mumbai, Singapore, Guangzhou,

	Source Name	Abbreviation	Link
1	Parallel Workloads Archive	PWA	http://www.cs.huji.ac.il/labs/parallel/workload/logs.html
2	Grid Workloads Archive	GWA	http://datamob.org/datasets/show/grid-workloads-archive
3	Grid Observatory	GOB	http://www.grid-observatory.org/
4	LANL Trace Data	LANL	http://institute.lanl.gov/data/tdata/
5	The Failure Trace Archive	FTA	http://fta.scem.uws.edu.au/index.php?n=Main.DataSets
6	Maui Scheduler Traces	MST	http://docs.adaptivecomputing.com/maui/16.1simulationoverview.php
7	NPACI JOBLOG Repository	NJP	http://www.cs.huji.ac.il/labs/parallel/workload/l_sdsc_sp2/
8	The computer failure data repository	CDFR	https://www.usenix.org/cfdr
9	The Internet Traffic Archive (Internet)	ITA	http://ita.ee.lbl.gov/index.html
10	The RIPE Data Repository (Internet)	RDR	https://labs.ripe.net/datarepository/data-sets

Table 6.1 Various archives and source of real traces of HPC jobs.

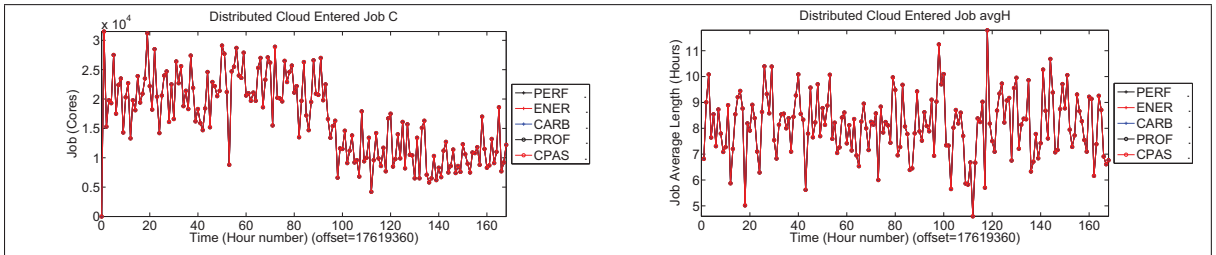
	Trace Name	Source	Length (Months)	CPUs	# Jobs	% Utilization
1	SDSC BLUE (Wang and Chu, 2009; de Assuncao *et al.*, 2009)	PWA	32	1,152	243,314	% 76.8
2	LLNL Thunder (Garg *et al.*, 2011; Etinski *et al.*, 2010)	PWA	5	4,008	121,039	% 86.7
3	Grid5000 (Iosup and Epema, 2011; Wu *et al.*, 2010)	GWA	30	2,500	951,000	% 10.4
4	EGEE Grid Trace (Rodero *et al.*, 2010; Lingrand *et al.*, 2010)	GOB	18	140,000	400,000/m	NA
5	NorduGrid (Caron *et al.*, 2010; Molfetas *et al.*, 2011)	GWA	21	2,000	781,000	% 69.8
6	GLOW (Rodero *et al.*, 2010; Iosup and Epema, 2011)	GWA	4	1,400	216,000	% 11.8

Table 6.2 Some of the most cited HPC traced in the literature.

Fukushima, and Sydney. The associated data centers will be referred as DC1 to DC10, respectively.

Since the data centers are located in different regions, therefore their energy mix, energy price, and temperature are different from each other. The energy (electricity) mix data is produced based on real data collected from publicly available power mix data of Ontario province[1]. For the other states, having their power mix percentages, the actual power mix data is generated

[1]http://reports.ieso.ca/public/



(a) Total amount of cores requested by new jobs (b) Average default length of new jobs

Figure 6.1 HPC workload features

based on data of power plants in the Ontario to match their overall emission factors. The electricity mix for various regions across the world are listed in Table 6.3, and the electricity price is provided in Table 6.4. The information for these tables are collected from sources listed in Table 6.5. A simple peak hour pattern similar to the Ontario state is used for the states with electricity peak hours. In the Figures 6.2-a, greenness of these areas are presented, and in Figure 6.2-b, their electricity price peaks are illustrated.

City	Nuclear (%)	Cole (%)	Gas (%)	Hydro (%)	Wind (%)	Other (%)	Oil (%)
Los Angeles	11	52	26	6	0	5	0
Sao Paulo	3	3	6	80	0	4	4
Humburg	24	46	15	5	7	1	2
Cape town	2	67	2	1	0	9	19
Mumbai	2	69	10	14	2	0	3
Singapore	0	0	78	0	0	4	18
Guangzhou	2	79	1	17	0	0	1
Fukushima	23	27	26	8	0	3	13
Sydney	0	77	15	5	1	1	1

Table 6.3 Energy mix data

City	Electricity price (US dollars cents per kWh)
Los Angeles	19
Sao Paulo	34
Humburg	34
Cape town	8 - 16
Mumbai	8 - 12
Singapore	20
Guangzhou	7 - 11
Fukushima	20 - 24
Sydney	22 - 46

Table 6.4 Energy price data

Region	Link
Los Angeles	http://www.bls.gov/ro9/cpilosa_energy.htm
Los Angeles	http://en.wikipedia.org/wiki/Los_Angeles_Department_of_Water_and_Power
Torronto	http://reports.ieso.ca/public/
Sao Paulo	http://en.wikipedia.org/wiki/Electricity_generation
Humburg	http://en.wikipedia.org/wiki/Electricity_pricing
Cape town	http://en.wikipedia.org/wiki/Energy_in_South_Africa
Singapore	http://www.ema.gov.sg/media/files/publications/EMA_SES_2012_Final.pdf
India	http://www.powerexindia.com/PXIL/
Others	International Energy Agency - Key World Energy Statistics (2009) http://www.iea.org/stats
Others	http://en.wikipedia.org/wiki/Electricity_generation
Others	http://en.wikipedia.org/wiki/Electricity_pricing

Table 6.5 Energy mix and price data sources

The carbon taxes are selected based on the information listed in the Table 6.6. It has been shown that the CO_2 taxation needs to be quite severe in order to influence toward more local sourcing (Global Commerce Initiative and Capgemini, 2008). Therefore, a virtual carbon tax of 30 US dollar cents per Kg of CO_2 is used in the virtual carbon tax scenarios.

Study	Carbon Tax		Industry	Country (region)
	$ / tCO_2	The currency of the study/ tCO_2		
Adamou *et al.* (2012)	42.58	31.04 €/ $tCO_2^{(a)}$	Car	Greece
Braathen (2012)	16.16	166.67 ZAR / $tCO_2^{(b)}$	Car	South Africa
Chen (2013)	5.60	34 CNY / $tCO_2^{(c)}$	Aggregated industry	China
Wang and Neumann (2009)	3.29-32.94	20-200 CNY / $tCO_2^{(d)}$	Aggregated industry	China
Chua and Nakano (2013)	796.4	1000 SGD / $tCO_2^{(e)}$	Car	Singapore
Gagoa *et al.* (2013)	253.8	185 €/ $tCO_2^{(f)}$	Electricity (Residential)	Spain
Gagoa *et al.* (2013)	34.30	25 €/ $tCO_2^{(f)}$	Electricity (Industry)	Spain
Gagoa *et al.* (2013)	480.2	350 €/ $tCO_2^{(f)}$	Transport (Gasoline)	Spain
Gemechu *et al.* (2013)	27.44	20 €/ tCO_2	Permit price	EU
Lundgren and Marklund (2012)	11.25	74 SEK / tCO_2	Mining	Sweden
Lundgren and Marklund (2012)	22.03	145 SEK / tCO_2	Food	Sweden
Lundgren and Marklund (2012)	19.00	125 SEK / tCO_2	Pulp/paper	Sweden
Medina (2013)	55.76	345 NOK / tCO_2	Aggregated industry	Norway
Pereira and Pereira (2013)	23.32	17 €/ tCO_2	Aggregated industry	Portugal
Xianqiang *et al.* (2013)	49.41	300 CNY / tCO_2	Transport (Gasoline)	China
Zimmermannová (2013)	20.58	15 €/ tCO_2	Energy	EU
CORNWELL and CREEDY (1996)	90.4	113 AUD / tCO_2	Aggregated industry	Australia
NERA Economic Consulting (2013)	20	20 $ / $tCO_2^{(g)}$	Aggregated industry	USA (California)

Table 6.6 Some carbon tax rates considered in the literature in various industries across the world.
$^{(a)}$ Assuming an emission rate of 145 gCO_2/Km, 15 €tax for every 1 gCO_2emission rate above than 100 gCO_2, a working range of $150,000$ Km. $^{(b)}$ Assuming an emission rate of 180 gCO_2/Km, 75 ZAR (South African currency) tax for every 1 gCO_2emission rate above than 120 gCO_2, a working range of $150,000$ Km. $^{(e)}$ Targeting to reach an Electrical Vehicle (EV) share of %10 in Singapore by 2020. $^{(f)}$ Implicit tax on carbon emissions of energy consumption. $^{(g)}$ 20$ has been marked as both fixed carbon tax and also the carbon tax that is required to have a %80 reduction in emissions in 2013.

The weather information regarding any of these locations are obtained from National Oceanic and Atmospheric Administration[2] databases. In Figure 6.2-c, the environment temperature is presented.

For the CPU specifications, similar to Garg *et al.* (2011) work, the parameters of the power consumption model (Equation (1.6)) are set to $\alpha = 0.004$ (kWh$/f^3$) and $\beta = 0.05$ (kWh). The minimum and maximum frequency of the CPUs are considered to be 1.5 and 3 GHz. The sales rate of the system is set to 6 cents per a core-hour @3GHz which is obtained based on the avarage sales rate of Amazon EC2.

All the experiments are done for one week period of time from 2nd January 2010 to 9th January 2010, and all the associated data regarding emission factors, electricity prices, and weather

[2]http://www.noaa.gov

temperatures are collected from this period of time. The "hour number"[3] indicated on the time axes of the graphs is referring to the same period of time. All the simulation is done in Matlab environment.

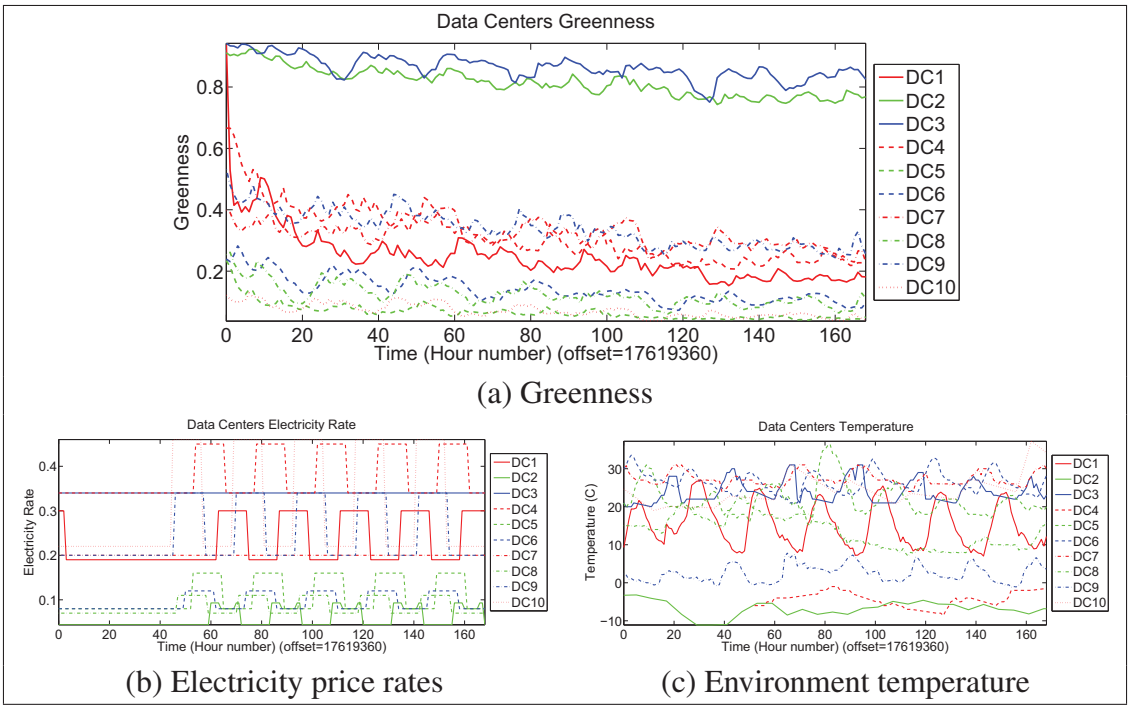


Figure 6.2 Data centers greenness, electricity price rates, and environment temperature

6.2.1.1 Comparing Algorithms

In Table 6.7, a list of algorithms with proper reference and a short description are presented which are used in this experiment to be compared with each other.

For the CPAS algorithm, it uses the Equation (4.3) as the profit model to calculate the total profit of the system. At the same time, it uses the Equation (4.5) to calculate the optimum frequency of the CPUs for each core-hour of the system variable space. It also uses this optimum frequency to calculate the maximum profit achievable in each core-hour and uses this information as a metric to guide the CPAS scheduler algorithm. A Pseudo code in Section 4.3 describe the functionality of the CPAS scheduler step by step.

[3]Please refer to Appendix I for definition.

Table 6.7 The comparison table among schedulers used in experimental setup.

Code name	Algorithm name	Main references	Description
PERF	Performance-based scheduler	Kim *et al.* (2003), Freund *et al.* (1998)	It uses a Percent Best MinMin scheduler to minimize the MCT metric.
ENER	Energy-based scheduler	Zhang *et al.* (2010)	It optimizes total system energy consumption to obtain the best frequencies for CPU cores.
CARB	Carbon-based scheduler	Garg *et al.* (2011)	It uses an optimal frequency for each type of CPU to minimize the energy consumption of jobs. It also uses MINMin scheduler to chose the best green servers.
PROF	Profit-based scheduler	Qureshi *et al.* (2009)	It uses a cost-aware request routing policy which is aware of variation of electricity price over time and location to minimize the cost.
CPAS	Carbon-Profit-Aware scheduler	This thesis	It optimizes the PpCHG metric to obtain the best frequencies for CPU cores. It also uses VCT to intensify the carbon footprint reduction.

6.2.2 CPA Scheduler Performance Study

This section will investigate the performance of the CPAS algorithm and will compare it with other state-of-the-art algorithms of the job schedulers. In Figure 6.3, profit of the system is presented for several algorithms. As shown in the figure, the CPAS algorithm has a better performance than other algorithms in terms of affording gain. The PERF algorithm has a better performance than the CARB algorithm when the system is fully utilized which is from hour zero to 120 in the Figure 6.3, and it has lower performance when the system is underutilized which is from hour 120 to 168. As it was illustrated in the Figure 6.1-a the amount of entered jobs reduces in hour 96 which switches the system from fully utilized to underutilized. The difference between hour 96 and hour 120 is due to the average deadline of the jobs which is 24 hours. The scheduler has in average 24 hours to schedule a job or it will fail. Therefore, even though the amount of jobs reduces in hour 96, they will remain in the system memory to be scheduled or failed until they reach their deadline. As it was projected in introduction and Section 2.1.3, the CARB algorithm works better in underutilized situations which is shown here.

In Figures 6.4 and 6.5 the energy consumption and average PUE of the system is presented. The energy consumption includes energy consumption of the IT equipments as well as support system. The average PUE shows the relation between energy consumption of the IT equip-

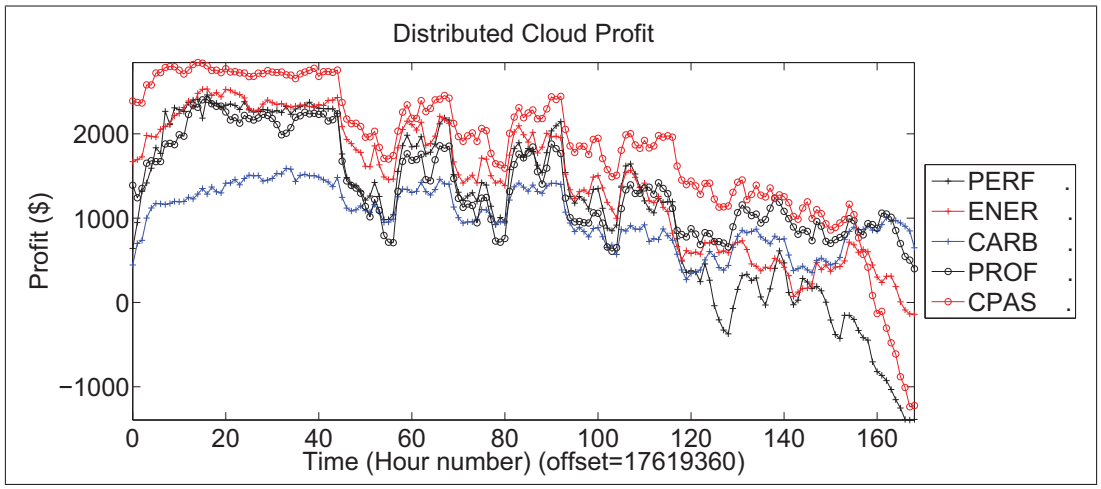


Figure 6.3 Geo-DisC profit

ments and support system which is represented by energy consumption of the cooling system. As it is shown in the Figure 6.4, the energy consumption of PERF algorithm is the highest due to its maximum CPU frequency, and energy consumption of CARB is the lowest due to its optimum CPU frequency. However, as it was shown in the Figure 6.3, even though the energy consumption of CPAS is higher than CARB algorithm, its profit is also higher which confirms the complex correlation relation between metrics of the system. Since the PERF algorithm uses the maximum frequency of the CPUs and fully utilizes the system, its sale and therefore profit is higher when there are enough jobs to fully utilize the system, but when the number of jobs reduces, it will loose the advantage of higher sale and profit. In addition, as it is shown in the Figure 6.5, the average PUE of PERF and CARB are in the same range and higher than CPAS algorithm. The reason for this observation is due to the fact that PERF and CARB algorithms do not have any mechanism in their decision process to consider the variations of the cooling system. On the other hand, CPAS algorithm has a mechanism to consider the cooling system indirectly. When the PUE is higher than its average value, the associated energy consumption and carbon footprint of the cooling system are also higher than their average value. Therefore, this situation lowers the profit of the system and since CPAS algorithm is a profit based algorithm it will avoid this situation by avoiding servers in areas with high PUE or by lowering the CPU frequency.

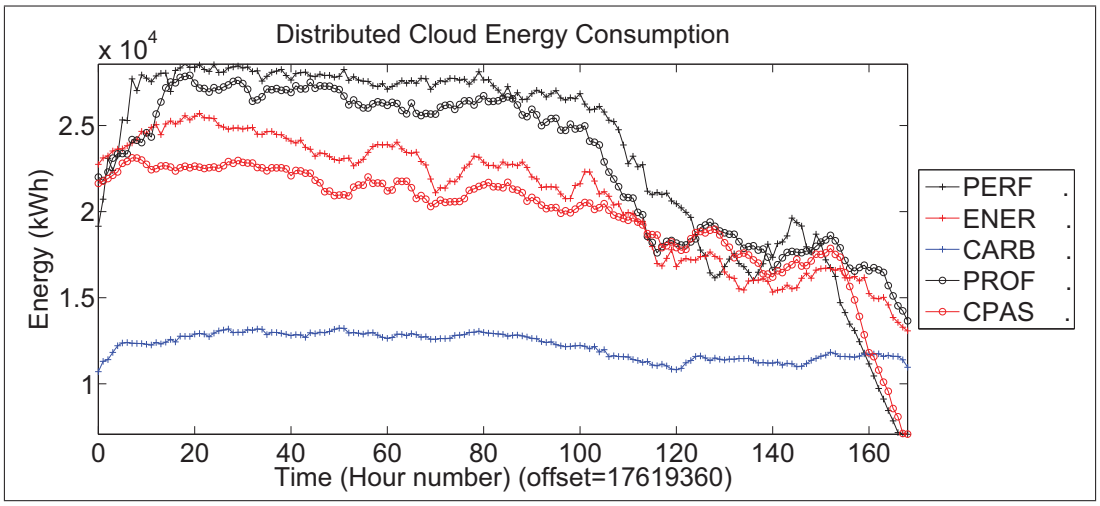


Figure 6.4 Geo-DisC energy consumption

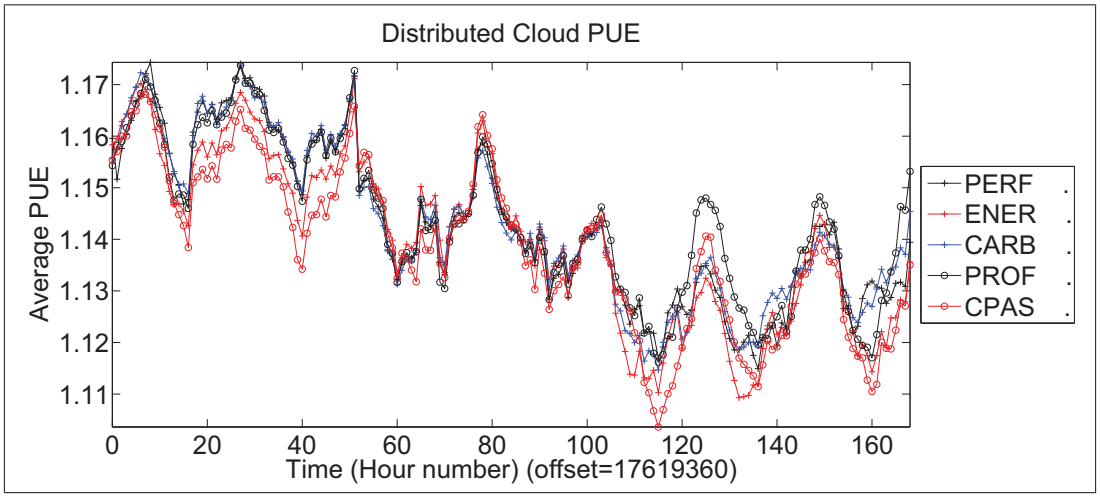


Figure 6.5 Geo-DisC average PUE

In Figures 6.6, the carbon footprint of the system is illustrated. This graph and profit graph together can show the success of the algorithms in terms of achieving both objectives of the this research: high profit and low carbon footprint. However, in the Figure 6.7, the greenness of CPAS is not at the top of the algorithms, even though it is comparable. As mentioned in the description of the CPAS algorithm, to boost the carbon reduction of this algorithm, virtual carbon tax needs to be considered. In the following sections, it is shown how virtual carbon tax can improve the greenness of CPAS with a small compromise in profit.

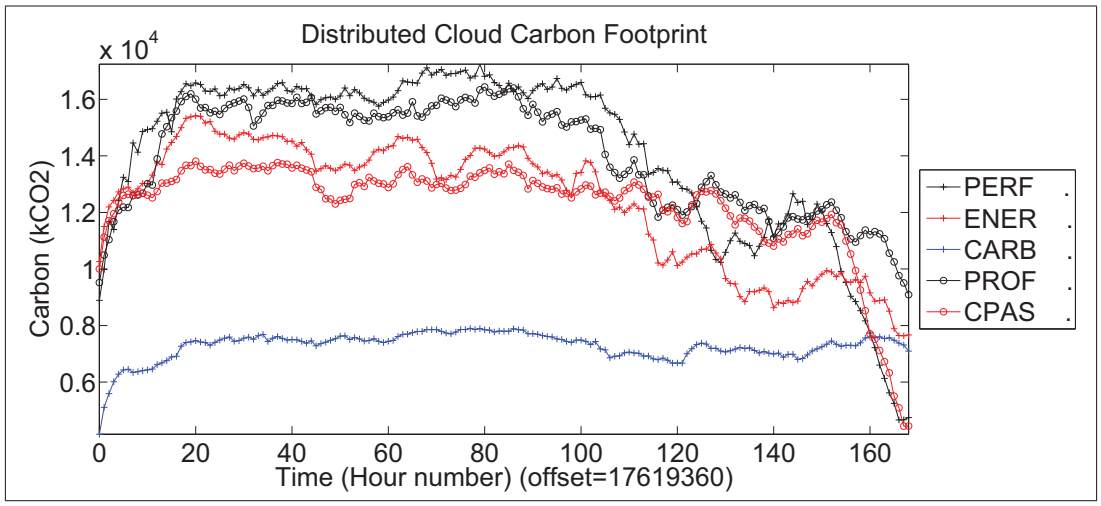


Figure 6.6 Geo-DisC carbon footprint

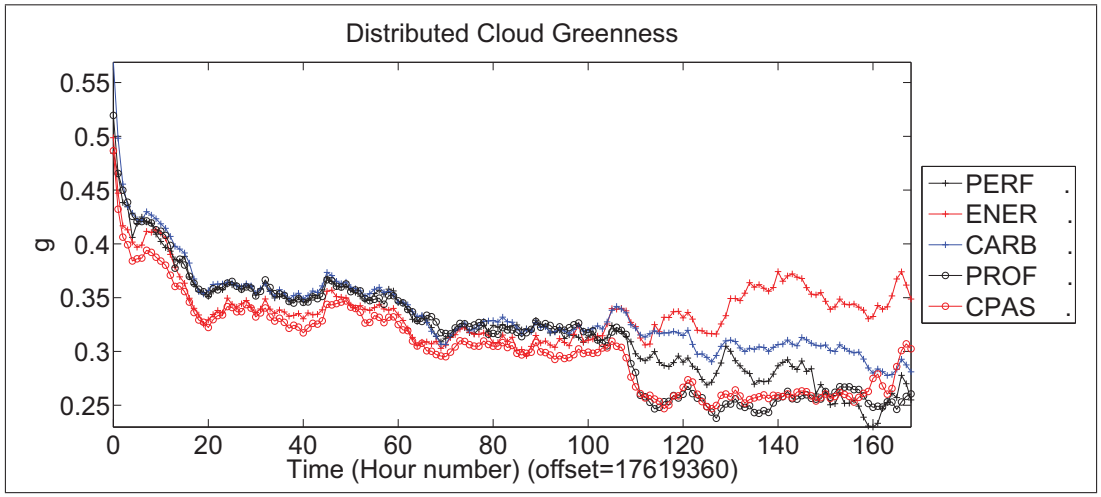


Figure 6.7 Geo-DisC greenness

As it was mentioned in the literature review and section 2.1.3, PERF and PROF algorithms use the maximum frequency of the CPUs while CARB calculates an optimum frequency for minimum energy consumption of the jobs. On the other hand, the CPAS algorithm chooses the optimum frequency of each job based on carbon tax, energy price, and sales rate parameters of the system. Moreover, the ENER algorithm uses a DVFS technique to adjust the frequency of jobs. Figure 6.8 present the variation of average frequency of the CPUs in CPAS algorithm.

As shown in the figure, the average frequency of CARB and PERF algorithm is predetermined and constant.

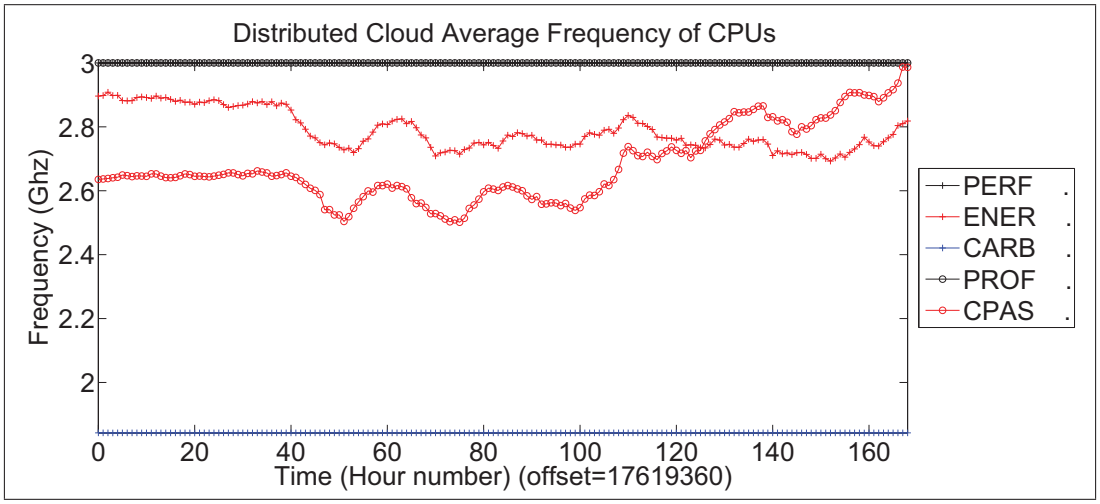


Figure 6.8 Geo-DisC average frequency

In the Figures 6.9, 6.10, and 6.11, a color coded sample of scheduled jobs on the servers is presented. The color code which is used here is fully described in Section 4.3.1. In this representation, each job is represented with a rectangular with a face color. Since there are 160000 CPU cores in this experiment, it is impossible to illustrate all of the cores in the schedule of jobs map, therefore, a pool of 160 cores (10 server) from each data center is selected to represent different data centers in this map. There are total of 1600 CPU cores presented in this map which are stacked from 1 to 1600 and are representing data center 1 to 10. The lowest cores are associated with the first data center and highest cores are associated with the last data center. Each scheduled job has a start time and a duration. The jobs with higher number of cores are thicker than the jobs with lower number of the cores.

These schedules need to be compared with the optimum profit map, which is defined in Section 4.1.3. The optimum profit map is illustrated in the Figure 6.12, and the associated color-code map is presented in Figure 6.13.

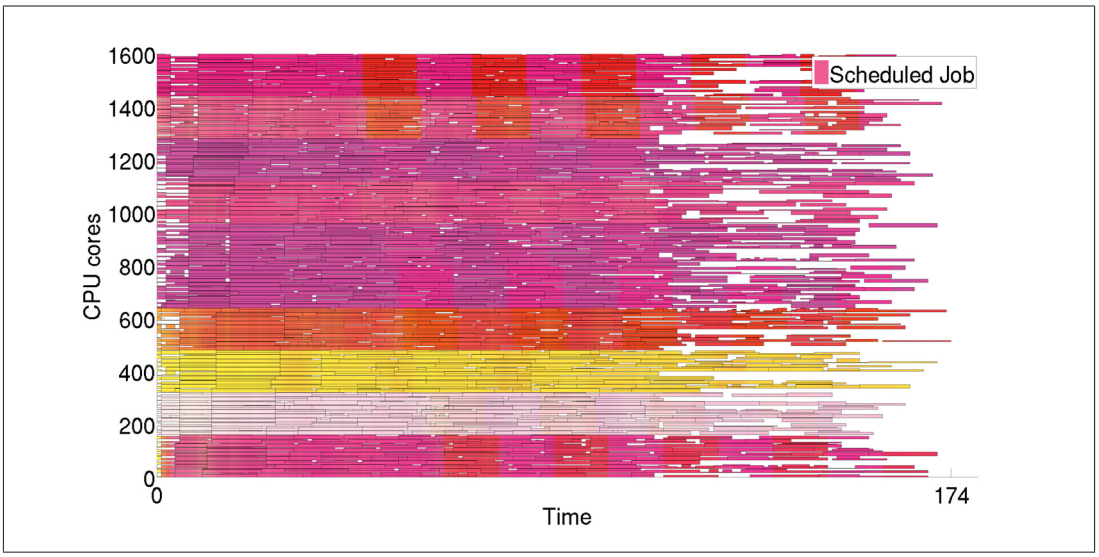


Figure 6.9 Scheduled jobs plot by PERF algorithm

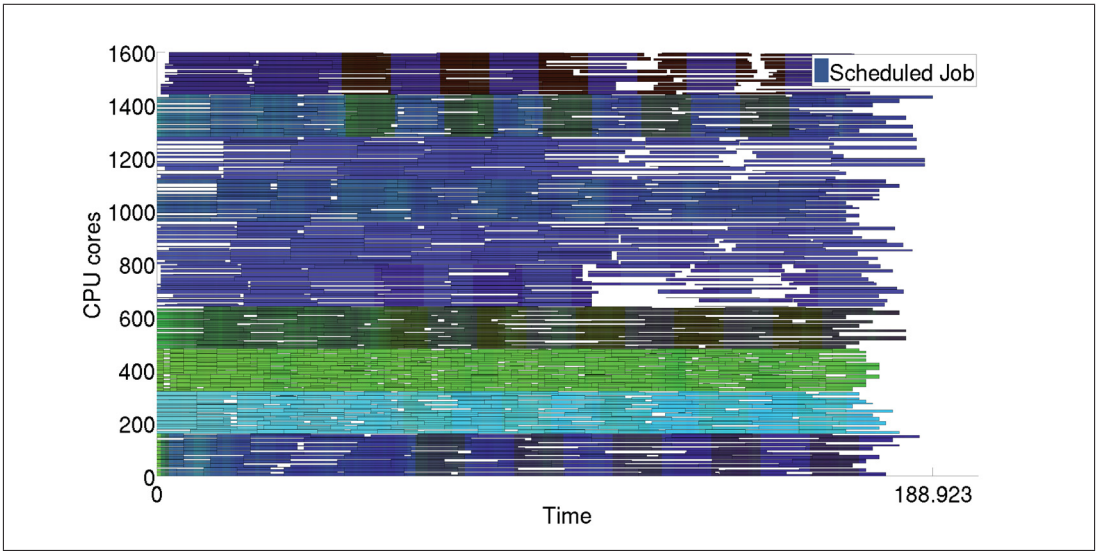


Figure 6.10 Scheduled jobs plot by CARB algorithm

The black dots in the color code illustrate the actual state of data centers in profit-frequency space. As it is shown, these points are around the optimum values (green dots), but they are not exactly on the optimum value. This is due to the fact that CPAS algorithm chooses one optimum frequency for each job regardless of the length of the job. Therefore a job which is optimum for profit in this hour may not be completely optimum in the coming hours. The other

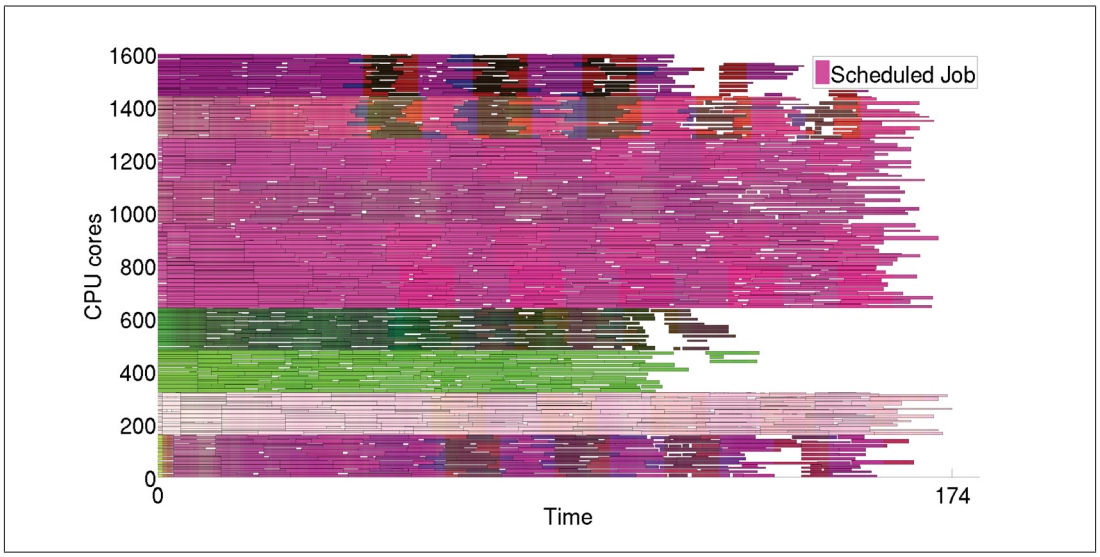


Figure 6.11 Scheduled jobs plot by CPAS algorithm

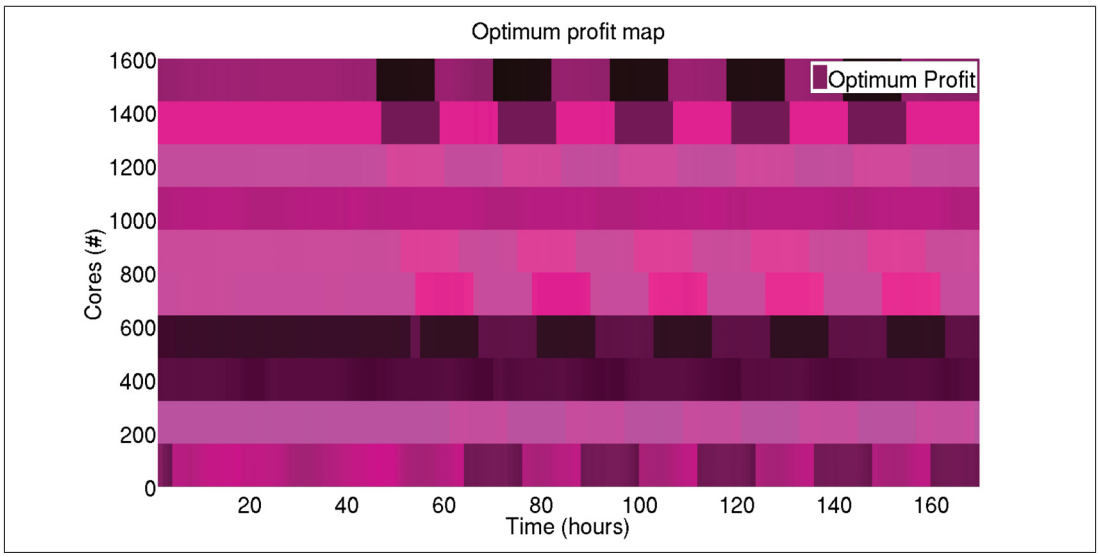


Figure 6.12 Optimum profit map

reason is the error margin of the predictions which is made for the parameters of the system. However, as shown in Figures 6.14, the state of CPAS is much better than the state of other algorithms.

From all above mentioned graphs, an average value is calculated for each operation hour of the system and the results are presented in the Table 6.8. As it is reported in this table, the

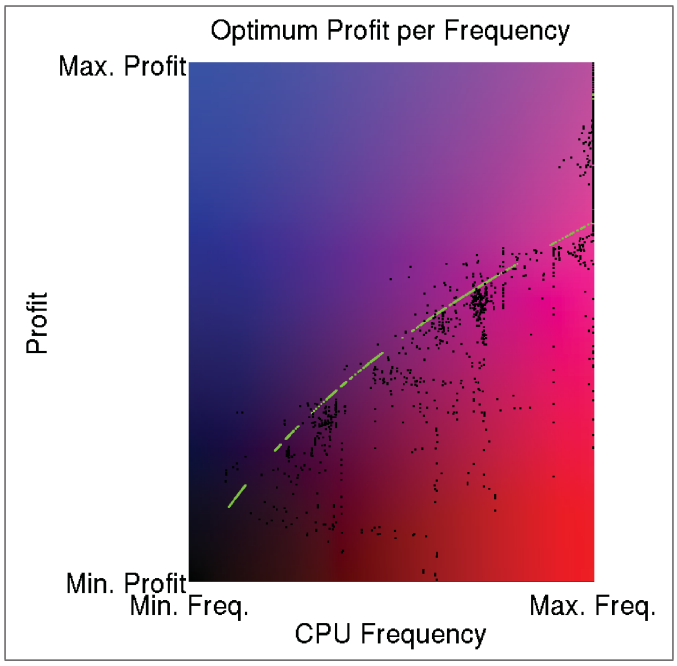


Figure 6.13 Profit per frequency for CPAS algorithms

CPAS algorithm has the best performance in profit than other algorithms. In addition, the CPAS algorithm has a better PUE factor than the other algorithms. However the greenness of the CPAS algorithm is not better than the other algorithms. The reason for these results is shown in the Sankey diagram of the cost and profit of the system in Figure 6.15. The carbon tax amount is so small that it cannot have an impact on the final results of the CPAS algorithm. Since carbon reduction is one of the goals of this thesis, to achieve both goals of this research, CPAS algorithm must be used with virtual carbon tax metric to guarantee achievement of both objectives of the thesis. The correspondent results are reported in Section 6.2.5.

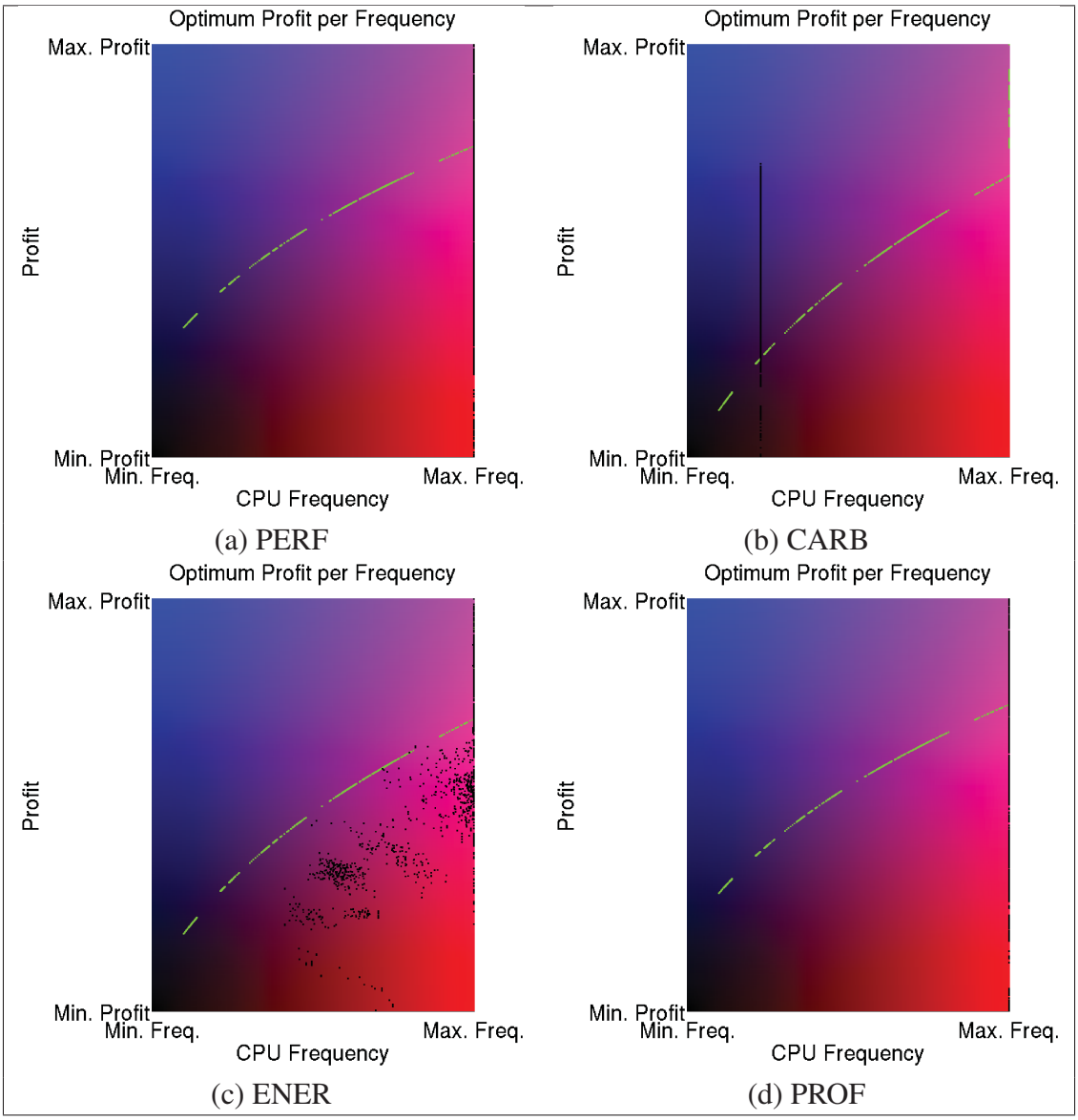


Figure 6.14 Profit per frequency for different algorithms

Table 6.8 The comparison table for performance study

Metric	PERF	ENER	CARB	PROF	CPAS
Virtual Carbon Cost ($)	**0**	**0**	**0**	**0**	**0**
Energy Consumption (kWh)	23473.18	20876.9081	**12196.6722**	23031.2002	19726.5915
Carbon Footprint (kgCO2)	14140.9784	12401.146	**7276.2217**	14010.2001	12270.0918
Greenness	0.32276	**0.34013**	0.33637	0.31685	0.30478
Total Cost ($)	6242.8024	5426.5465	**3810.1204**	5833.8086	4868.6053
Sale ($)	**7394.4852**	6864.7731	4820.8713	7206.8521	6706.4404
Profit ($)	1151.6828	1438.2266	1010.7509	1373.0435	**1837.8351**
Profit plus VCT ($)	1151.6828	1438.2266	1010.7509	1373.0435	**1837.8351**
Average Freq (Ghz)	**3**	2.7871	1.842	**3**	2.678
Scheduled Job (CHG)	**370010.0592**	346693.4911	248304.142	363928.4024	333789.9408
Failed Job (CHG)	**36237.8698**	59554.4379	153434.9112	42319.5266	72457.9882
PUE	1.1429	1.1403	1.1431	1.1438	**1.1386**
Running Jobs (CHG)	**369724.2604**	343238.655	241043.5651	360342.6036	335322.0196
Slacks (Cores)	36758.5799	37084.0237	**29141.4201**	39885.7988	33484.0237
Execution Time (sec)	64.5068	541.2964	**64.3541**	477.8096	764.1728

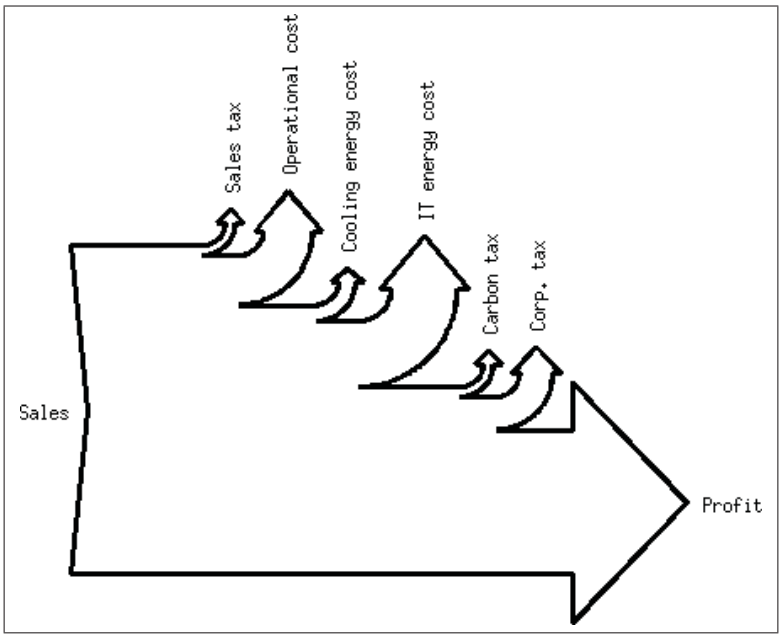


Figure 6.15 Sankey diagram of the cost and profit of the system

6.2.3 Seasonal Energy-Variations Study

The goal of this study is to show how energy variations in four different seasons affect the performance of the algorithms. In this study, two algorithms are tested on the same network for an interval of one week. In Figure 6.16-a carbon footprint of a typical Geo-DisC is presented. As it is shown, the carbon footprint of system is higher in January and July which is correct due to higher energy demand in summer and winter, which has a direct effect on the power mix. Respectively, in Figures 6.16-b, 6.16-c, and 6.16-d, energy consumption, profit, and greenness of the system is presented. The results of this experiment are summarized in Table 6.9.

For the similar system in Figure 6.17, the energy consumption is presented under the CPAS scheduler. Comparing Figures 6.17 and 6.16-b, it shows that carbon sensitive algorithms will respond to energy mix variations over the time and the scheduling is different in each different month, but for the non-sensitive algorithms to carbon and energy price, the energy consumption of IT equipments in different seasons is not impacted. The small variation of the total energy consumption in Figures 6.16-b is due to variations of cooling system power consump-

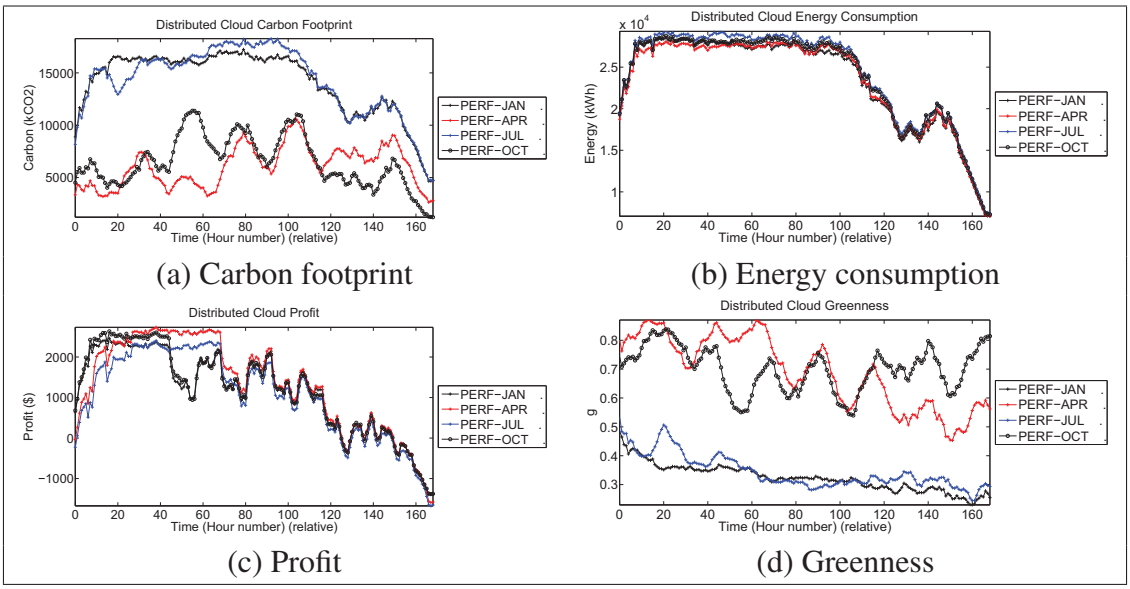


Figure 6.16 Geo-DisC in different seasons under PERF algorithm

Table 6.9 The comparison table for seasonal study of PERF algorithm

Metric	PERF-JAN	PERF-APR	PERF-JUL	PERF-OCT
Virtual Carbon Cost ($)	**0**	**0**	**0**	**0**
Energy Consumption (kWh)	23473.18	**23458.7961**	24399.7472	23969.3682
Carbon Footprint (kgCO2)	14140.9784	**6053.3711**	14270.8855	6525.0794
Greenness	0.32276	0.69204	0.34423	**0.70078**
Total Cost ($)	6242.8024	**6067.6499**	6336.215	6197.7281
Sale ($)	**7394.4852**	**7394.4852**	**7394.4852**	**7394.4852**
Profit ($)	1151.6828	**1326.8353**	1058.2702	1196.7571
Profit plus VCT ($)	1151.6828	**1326.8353**	1058.2702	1196.7571
Average Freq (Ghz)	**3**	**3**	**3**	**3**
Scheduled Job (CHG)	**370010.0592**	**370010.0592**	**370010.0592**	**370010.0592**
Failed Job (CHG)	**36237.8698**	**36237.8698**	**36237.8698**	**36237.8698**
PUE	1.1429	**1.142**	1.187	1.1659
Running Jobs (CHG)	**369724.2604**	**369724.2604**	**369724.2604**	**369724.2604**
Slacks (Cores)	**36758.5799**	**36758.5799**	**36758.5799**	**36758.5799**
Execution Time (sec)	65.0973	**64.6909**	65.1331	65.0861

tion caused by variations of temperature in different seasons. The results of this experiment are summarized in Table 6.10.

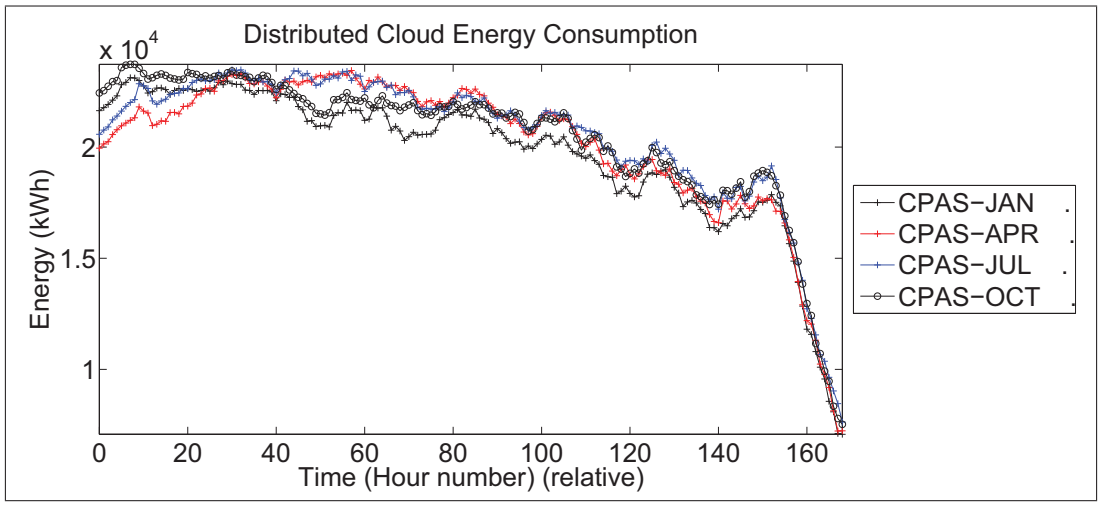


Figure 6.17 Geo-DisC energy consumption in different seasons under CPAS algorithm

Table 6.10 The comparison table for seasonal study of CPAS algorithm

Metric	CPAS-JAN	CPAS-APR	CPAS-JUL	CPAS-OCT
Virtual Carbon Cost ($)	**0**	**0**	**0**	**0**
Energy Consumption (kWh)	**19726.5915**	20214.9937	20577.1211	20479.3243
Carbon Footprint (kgCO2)	12270.0918	**5435.5851**	12355.229	5597.8182
Greenness	0.30478	0.68841	0.32981	**0.69866**
Total Cost ($)	**4868.6053**	4869.2903	4933.2713	4926.6047
Sale ($)	6706.4404	**6782.7954**	6679.1385	6762.1583
Profit ($)	1837.8351	**1913.505**	1745.8672	1835.5535
Profit plus VCT ($)	1837.8351	**1913.505**	1745.8672	1835.5535
Average Freq (Ghz)	2.678	**2.7105**	2.6616	2.6889
Scheduled Job (CHG)	333789.9408	**337732.5444**	332501.1834	336363.9053
Failed Job (CHG)	72457.9882	**68515.3846**	73746.7456	69884.0237
PUE	**1.1386**	1.1471	1.1947	1.1742
Running Jobs (CHG)	335322.0196	**339139.7684**	333956.9257	338107.9132
Slacks (Cores)	33484.0237	33563.9053	33162.1302	**32936.0947**
Execution Time (sec)	**757.6114**	758.1474	792.7486	781.4408

6.2.4 Cooling System Study

Figure 6.18 compares the carbon footprint of the NDC when electricity consumption related to the cooling system is included to that which ignores it. As can be seen, ignoring the cooling-related consumption introduces a big error in the footprint calculations that may misguide the NDC manager/controller in its decisions to displace the load and jobs flow to different data centers in order to reduce the overall footprint. In particular, the difference in carbon footprint is as high as 4000 kgCO_2 per hour at some moments. Furthermore, it is obvious from the figure that when the NDC manager takes into account the cooling-related consumption and also consider optimizing the placement in order to reduce the overall footprint, there is a

considerable reduction compared to the case without any optimization. To be precise, there is up to 2000 $kgCO_2$ per hour (almost half of the error if cooling system is ignored). This shows not only considering the dynamic behaviour of the cooling system in the calculations is essential in any modeling and management of an NDC, a cooling-aware manager can achieve considerable footprint reduction by choosing proper data centers at each moment of operation.

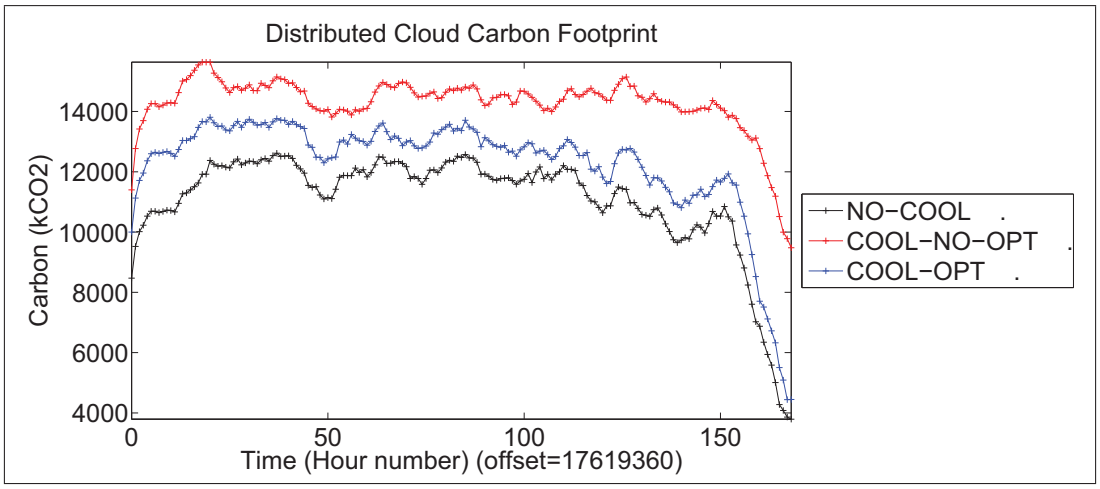


Figure 6.18 Geo-DisC carbon footprint (cooling system study)

A similar behaviour can be seen from energy consumption of the whole system, shown in Figure 6.19-a, with bigger impact of optimization on reduction of energy consumption. It can be seen that the consumption of the optimized operation almost reaches that of the NDC with ignored cooling system. This is reflected directly in the profit profiles, shown in Figure 6.19-b, where the optimized operation considerably achieves higher profit compared to not-optimized operation. The black curve, which corresponds to the case where the cooling system is ignored, is not realistic because the cooling system consumption should be considered before calculating the profit.

Finally, the average frequency of CPUs/cores is shown in Figure 6.19-c. The average-per-hour values of the metrics in this experiment are summarized in Table 6.11.

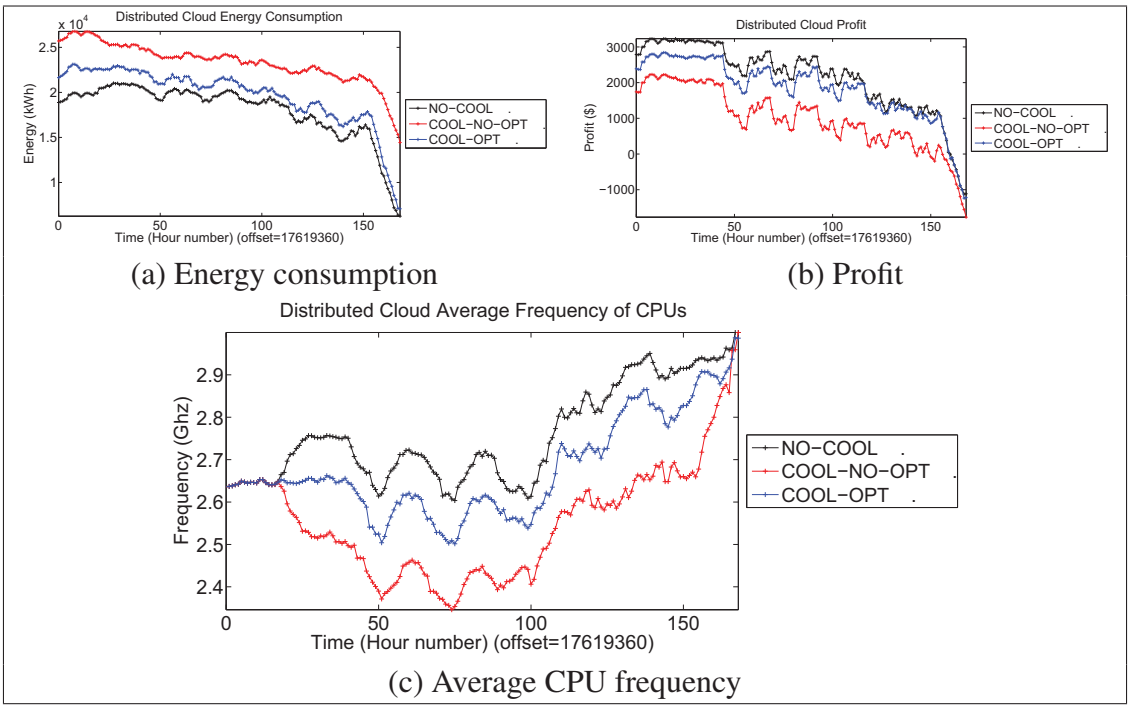


Figure 6.19 Geo-DisC under CPAS algorithm with different cooling strategies

Table 6.11 The comparison table for cooling study

Metric	NO-COOL	COOL-NO-OPT	COOL-OPT
Virtual Carbon Cost ($)	**0**	**0**	**0**
Energy Consumption (kWh)	18066.5576	23353.6504	**19726.5915**
Carbon Footprint (kgCO2)	11060.4425	14266.0074	**12270.0918**
Greenness	0.31691	**0.31823**	0.30478
Total Cost ($)	4703.366	5414.6661	**4868.6053**
Sale ($)	6851.2894	6415.4221	**6706.4404**
Profit ($)	2147.9234	1000.756	**1837.8351**
Profit plus VCT ($)	2147.9234	1000.756	**1837.8351**
Average Freq (Ghz)	2.7573	2.5538	**2.678**
Scheduled Job (CHG)	341105.3254	319874.5562	**333789.9408**
Failed Job (CHG)	65142.6036	86373.3728	**72457.9882**
PUE	1	1.4543	**1.1386**
Running Jobs (CHG)	342564.4716	320771.1062	**335322.0196**
Slacks (Cores)	34475.1479	**33232.5444**	33484.0237
Execution Time (sec)	755.8809	**728.375**	747.0297

6.2.5 Virtual Carbon Tax Study

In Section 6.2.2, it was shown how CPAS algorithm has the highest profit. although CPAS is also carbon sensitive, but in many states there is no carbon tax in place right now and the CPAS algorithm in these states act as a pure profit maximizing algorithm. In order to reduce the environmental impacts of the system in these states voluntarily, the VCT was defined in previous chapters. Here, the effect of the VCT on the CPAS algorithm and other algorithms

are evaluated. The rate of the VCT used here is 30 cents per kilogram carbon emissions and is the same for all the data centers.

In Figures 6.20-a, 6.20-b, 6.20-c, 6.20-d, 6.20-e, and 6.20-f, Geo-DisC profit, real profit, carbon footprint, energy consumption, greenness, and virtual carbon tax cost are presented, respectively.

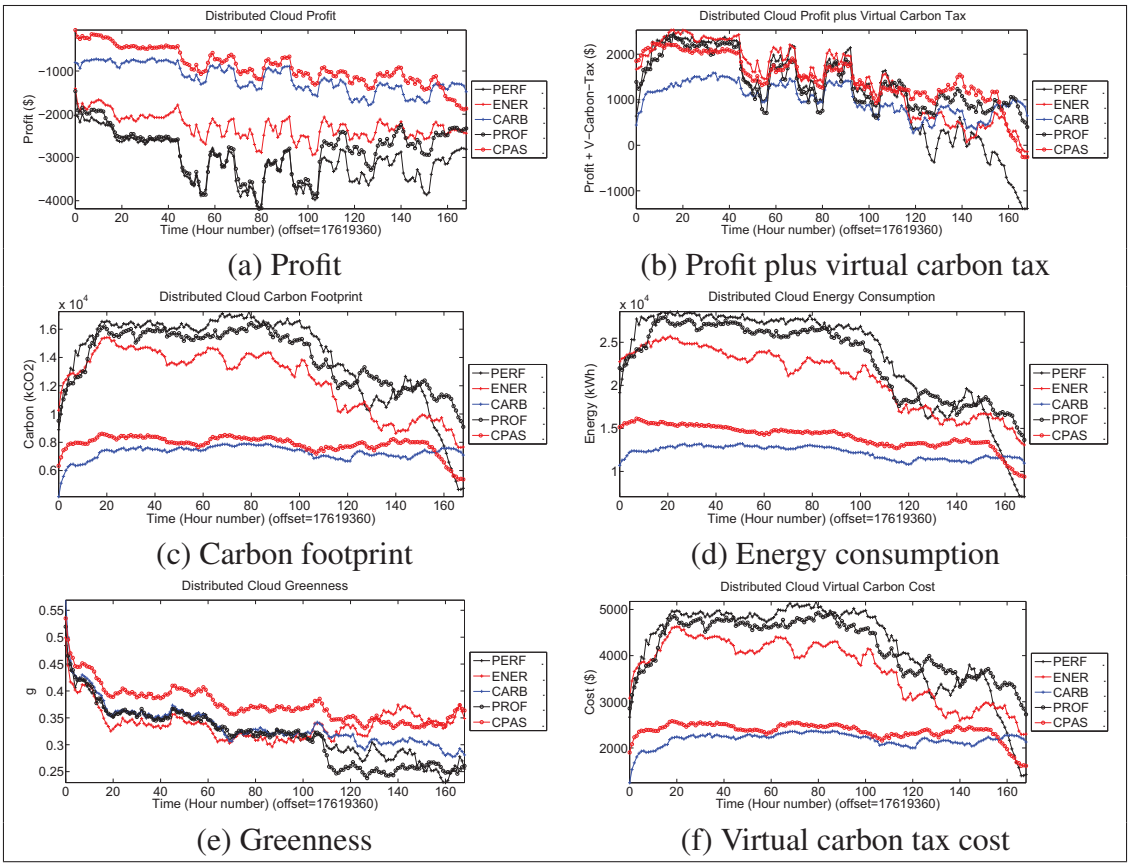


Figure 6.20 Geo-DisC under different algorithm with utilization of virtual carbon tax

As it is shown in the Figure 6.20-a the system is not profitable under high rate of the VCT, but as it is mentioned before, the real profit of the system is the combination of profit and cost of the VCT which is reported in the Figure 6.20-b which is totally profitable. The scenario with CPSA algorithm and high VCT is not only profitable, but also it has the greenest values among all other algorithms with different carbon rates. Area view of costs, profit, and sales of system

is presented in Figures 6.21, and the average-per-hour values of metrics in this experiment are summarized in Table 6.12.

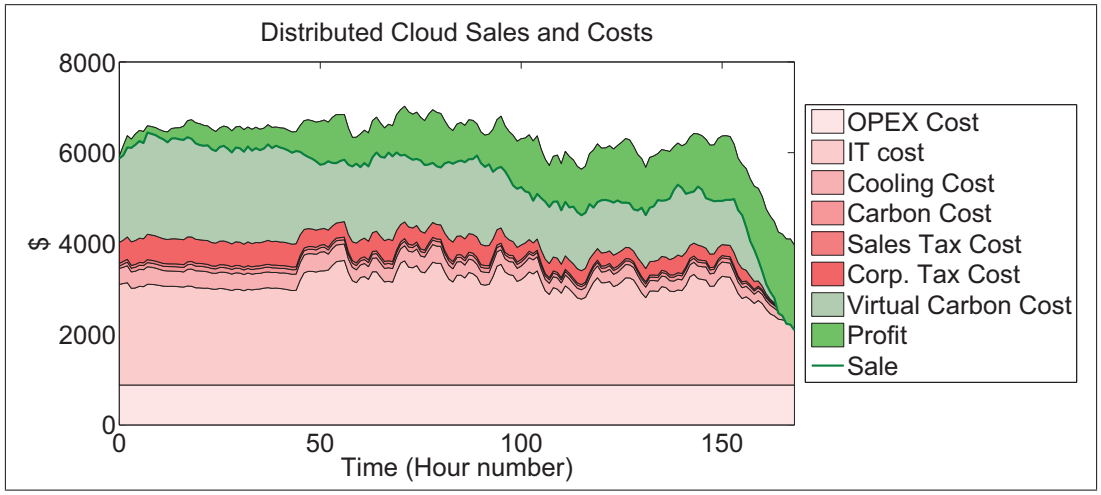


Figure 6.21 Geo-DisC sale, costs, and profit for high virtual-carbon-tax scenario and scheduled by CPAS

Table 6.12 The comparison table for virtual carbon tax study

Metric	PERF	ENER	CARB	PROF	CPAS
Virtual Carbon Cost ($)	**4242.2935**	3720.3438	2182.8665	4203.06	2362.6167
Energy Consumption (kWh)	23473.18	20876.9081	**12196.6722**	23031.2002	14060.338
Carbon Footprint (kgCO2)	14140.9784	12401.146	**7276.2217**	14010.2001	7875.3891
Greenness	0.32276	0.34013	0.33637	0.31685	**0.37558**
Total Cost ($)	10485.0959	9146.8903	**5992.9869**	10036.8686	6280.9617
Sale ($)	**7394.4852**	6864.7731	4820.8713	7206.8521	5388.1931
Profit ($)	-3090.6107	-2282.1172	-1172.1156	-2830.0166	**-892.7686**
Profit plus VCT ($)	1151.6828	1438.2266	1010.7509	1373.0435	**1469.8481**
Average Freq (Ghz)	**3**	2.7871	1.842	**3**	2.0528
Scheduled Job (CHG)	**370010.0592**	346693.4911	248304.142	363928.4024	270482.8402
Failed Job (CHG)	**36237.8698**	59554.4379	153434.9112	42319.5266	134460.355
PUE	1.1429	1.1403	1.1431	1.1438	**1.1295**
Running Jobs (CHG)	**369724.2604**	343238.655	241043.5651	360342.6036	269409.653
Slacks (Cores)	36758.5799	37084.0237	29141.4201	39885.7988	**28280.4734**
Execution Time (sec)	64.5961	529.1303	**64.1783**	486.7963	680.0952

Similar to Sankey diagram presented in the performance study of the CPAS algorithm, a Sankey diagram of the cost and profit of the system is presented in Figure 6.22. As shown in the figure, the total amount of carbon tax plus virtual carbon tax is significant and the CPAS algorithm is forced to reduce the carbon footprint of the system in order to maximize the profit. However, as it is shown in the diagram, the amount of VCT is aggregated with profit of the system and create the real profit of the system. Therefore, it is expected from the introduction of large

amounts of VCT to the system to decrease the carbon footprint, but it is not expected that VCT has the same impact on the real profit of the system. As it was observed in the performance study and here, the VCT reduces the carbon footprint of the system but also reduces the profit of the system. As it was mentioned before this is the trade-off between these two objectives of the system. Here the question remains that what is the best balance for these two objectives. Next section will investigate this question.

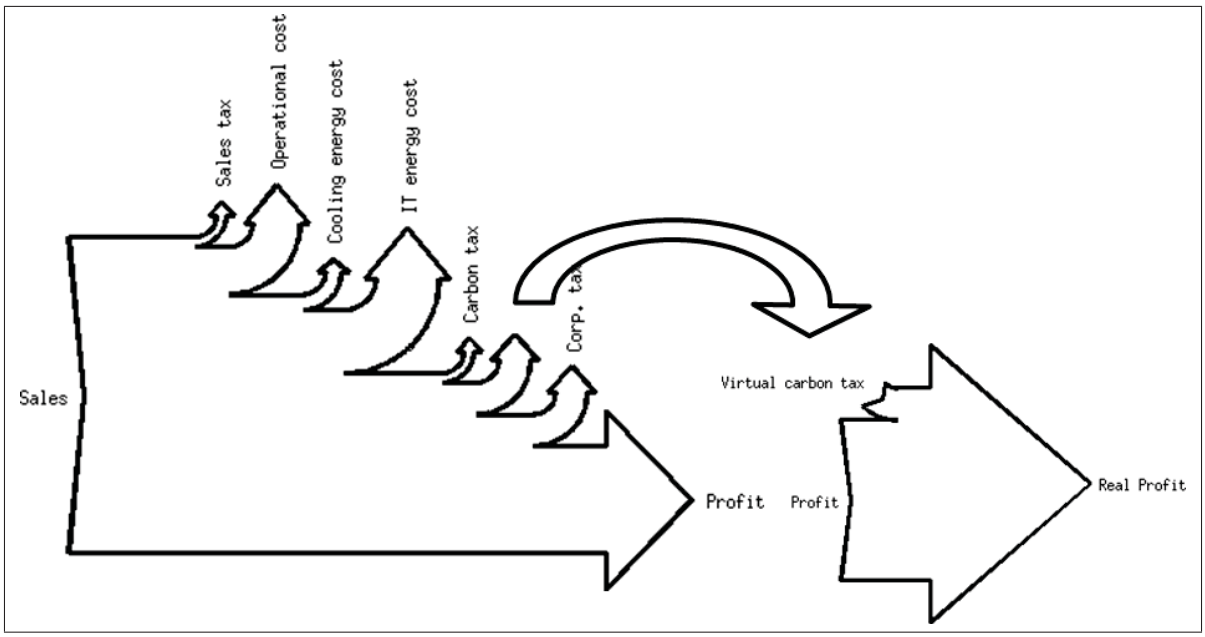


Figure 6.22 Sankey diagram of the cost and profit of the system with VCT

6.2.5.1 Carbon-Profit Trade-Off in CPAS with VCT

In the previous section, the effect of VCT was studied in different algorithms. Here, the effect of VCT will be studied on CPAS algorithm with different rates of VCT. In Figures 6.23-a, 6.23-b the profit and profit+VCT are depicted. As is shown, with introduction of carbon tax to the system, profit decreases a bit. However, the decrease in profit is much higher with introduction of VCT to the system. While the profit is significantly decreases after application of VCT on the system, the real profit of the system (profit+VCT) is in a closer range than profit.

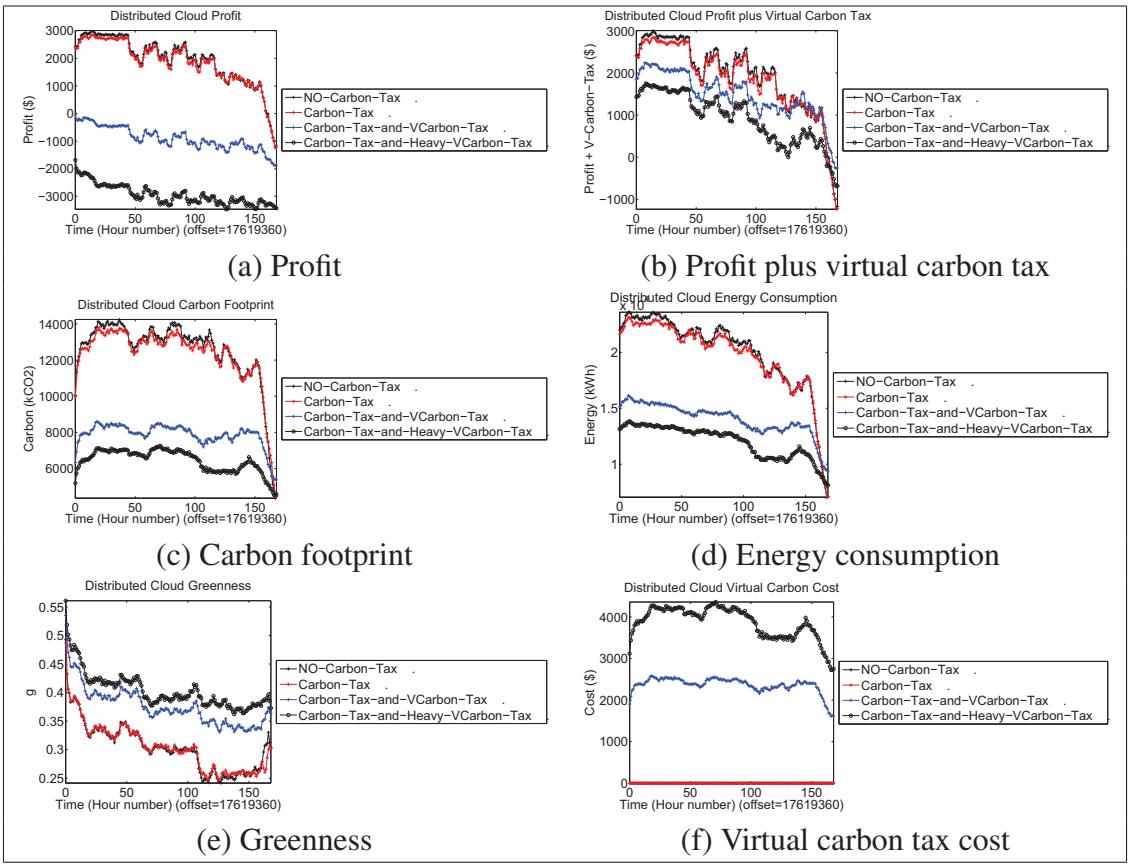


Figure 6.23 Geo-DisC under CPAS algorithm with different virtual carbon taxes

Although the real profit of system is decreasing with an increase in VCT, its impact on energy consumption and carbon footprint is significant as it is shown in Figures 6.23-d, 6.23-c. Not only the VCT reduces both energy consumption and carbon footprint of the system, its impact on carbon is relatively higher than energy as it is shown in Figure 6.23-e with a higher greenness. The effect of VCT is not limitless and with increase of VCT its effect decreases. As it is shown in the energy consumption and carbon footprint figures, with heavy VCT rates, the decrease in profit is significant while the decrease in carbon is not as much as moderated VCT rates. Therefore, heavy VCT rates may not be justifiable and moderate rates need to be used. The real tradeoff in a system with moderate VCT is between profit and carbon reduction. The system under moderate VCT rates looses some profit, but the reduction in carbon is relatively higher than this reduction in profit.

In figure 6.24 the areas of cost and profit and VCT are depicted in different scenarios, and the average-per-hour values of metrics in this experiment are summarized in Table 6.13.

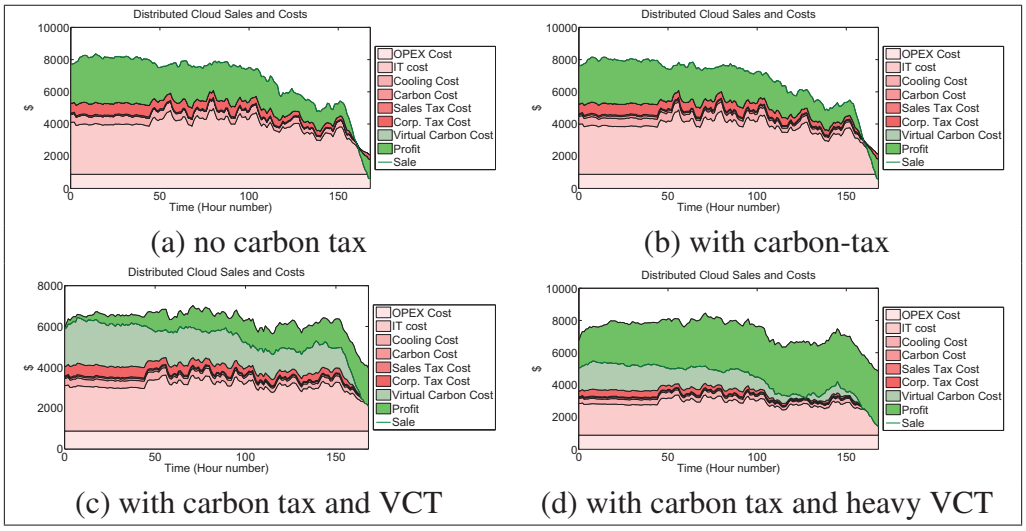


Figure 6.24 Geo-DisC sale, costs, and profit (scheduled by CPAS)

Table 6.13 The comparison table for virtual carbon tax study of CPAS algorithm

Metric	NO-CT	CT	CT-and-VCT	CT-and-HVCT
Virtual Carbon Cost ($)	0	0	2362.6167	**3861.8998**
Energy Consumption (kWh)	20055.6944	19726.5915	14060.338	**12035.4644**
Carbon Footprint (kgCO2)	12496.3835	12270.0918	7875.3891	**6436.4997**
Greenness	0.30392	0.30478	0.37558	**0.40344**
Total Cost ($)	**4862.9008**	4868.6053	6280.9617	7337.3007
Sale ($)	**6789.5656**	6706.4404	5388.1931	4390.629
Profit ($)	**1926.6649**	1837.8351	-892.7686	-2946.6718
Profit plus VCT ($)	**1926.6649**	1837.8351	1469.8481	915.228
Average Freq (Ghz)	**2.7116**	2.678	2.0528	1.7889
Scheduled Job (CHG)	**337997.0414**	333789.9408	270482.8402	222578.6982
Failed Job (CHG)	**68250.8876**	72457.9882	134460.355	176652.071
PUE	1.1376	1.1386	1.1295	**1.1258**
Running Jobs (CHG)	**339478.2817**	335322.0196	269409.653	219531.4478
Slacks (Cores)	33479.2899	33484.0237	**28280.4734**	36345.5621
Execution Time (sec)	748.0394	756.8532	668.0682	**584.9671**

To have a full picture of the the trade-off between carbon and profit, the experiment is done for a range of the VCT values, and the results are provided in the Figure 6.25. As shown in the figure and it was also mentioned earlier, with introduction of smaller amount of the VCT to the system, the profit does not reduced as much as carbon. It is almost like a flat line for small VCT amount. However, with increase in the amount of the VCT, the profit decrease with a much higher rate. This graph gives the business owners a clear picture of the trade-off between

profit and carbon, and it will be their decision to choose one of the many solutions provided here. According to this research, all these solutions are valid, but for example introduction of 15 cents of VCT per kilogram of carbon emissions results in not much decrease in profit with about two tones of carbon reduction in carbon footprint per hour. This results can be used as a self-motivating approach for business owners to take more carbon reduction measures while keeping their level of profit untouched or with a minimum impact. If this amount of carbon reduction was the result of real carbon tax, the businesses cannot survive (refer to the profit row of the Table 6.13).

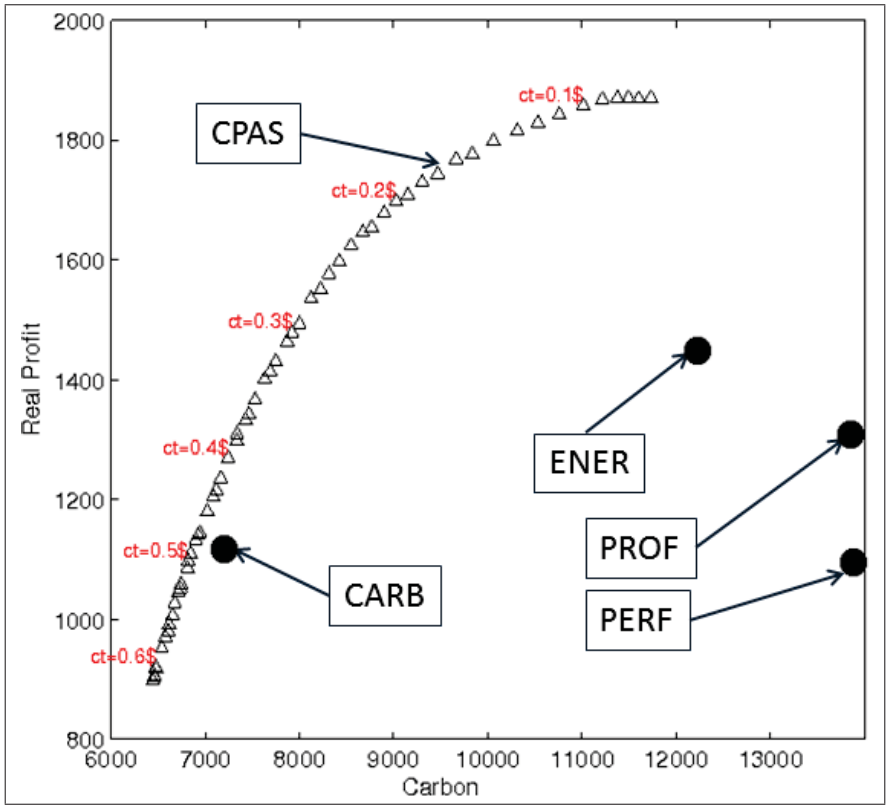


Figure 6.25 Carbon-profit trade-off per hour (ct represent the amount of VCT applied)

6.2.5.2 Study of CPA Scheduler based on Virtual GHG-INT Equivalent Carbon Tax

In previous section, it was observed that how introduction of virtual carbon tax can direct the scheduler towards lower carbon emissions. The same technique can be used for other metrics of

interest. According to the GHG intensity (GHGINT) indicator,[4] carbon emissions in different regions should not be evaluated only based on the amount of the emissions, but the amount of production achieved should be also considered. The GHGINT is not supported by all regions. To be precise, there are three major indicators proposed in this direction, and we will consider them in this thesis:

a. The GHG intensity (GHGINT) indicator (Jotzo and Pezzey, 2005):

$$\text{GHGINT} = \frac{\text{GHG}_{\text{total}}}{\text{GDP}} \tag{6.1}$$

where GDP stands for Purchasing Power Parity GDP (GDP (PPP)) (Farrahi Moghaddam *et al.*, 2013; The World Bank Group, 2011).

b. The GHG emissions per capita (GHGpCapita) indicator:

$$\text{GHGpCapita} = \frac{\text{GHG}_{\text{total}}}{\text{Population}} \tag{6.2}$$

where Population is the population in 1990 as the global baseline year of GHG emission reduction efforts (as proposed in Farrahi Moghaddam *et al.* (2013)).

c. The modified GHG intensity (MGHGINT) indicator (Farrahi Moghaddam *et al.*, 2013):

$$\text{MGHGINT} = \frac{\text{GHG}_{\text{total}}}{\text{IHDIGDP}} \tag{6.3}$$

where IHDIGDP is the IHDI-adjusted GDP (Farrahi Moghaddam *et al.*, 2013), and IHDI is inequality-adjusted human development index (Alkire and Foster, 2010).

The GHG emissions in a region with a high GDP is less intense (has a less GHGINT indicator) compared to a region with a lower GDP value. In contrast, the same amount of emissions would have a higher GHGpCapita indicator in a region with a lower population compared to a region with a higher population. These two indicators can be very much in contradiction,

[4] Refer to Appendix II for more details.

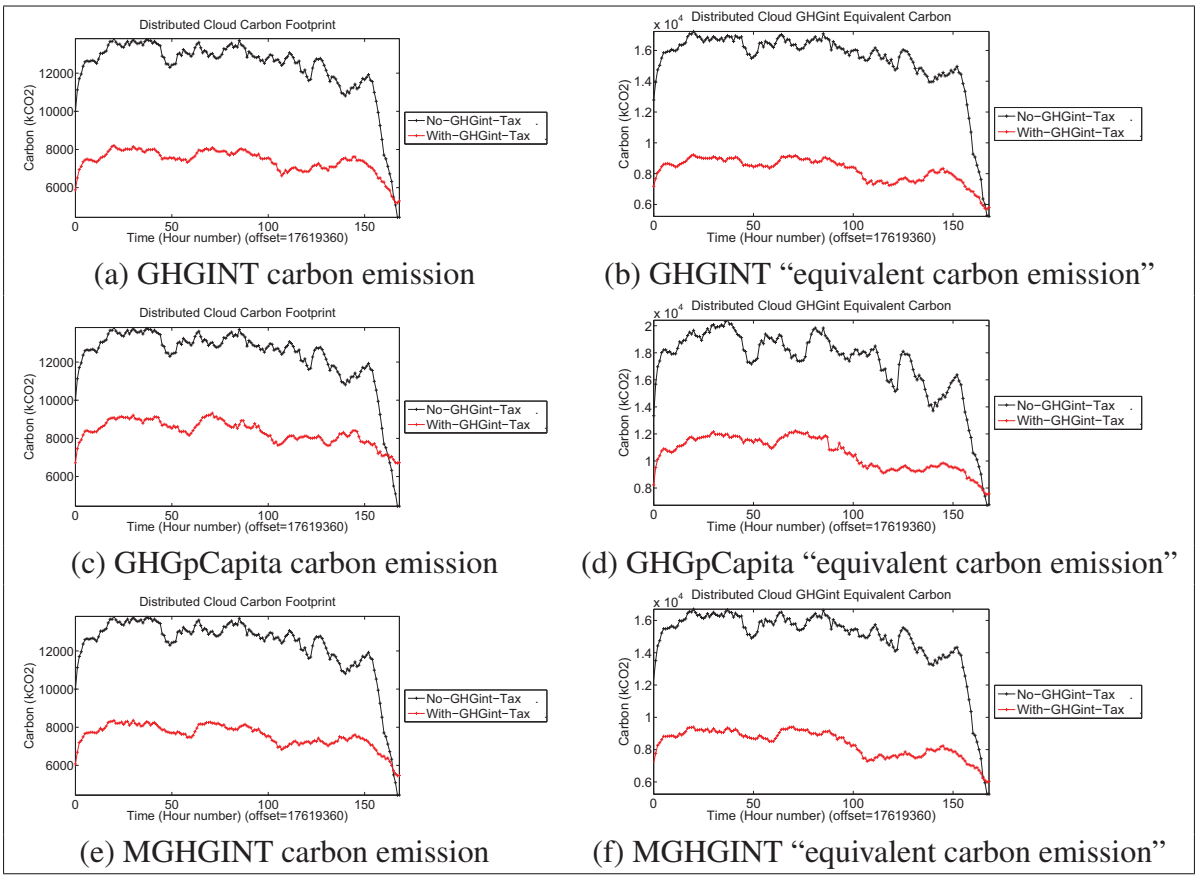
and this has been the main reason for lack of global agreement on a universal and unique indicator, and justifies the necessity of the third indicator, i.e., the MGHGINT indicator, which provides a fair and universal assessment of GHG emissions of all regions. Regardless of which indicator is chosen, the virtual carbon tax can be used to minimize it. In the following, the Geo-DisC system is tested considering all these three indicators. For example, in Equation 6.4, an "equivalent carbon emission" metric is introduced in order to measure the carbon footprint of the system with respect to the GHGINT indicator.

$$C_{\text{eq}} = C * \frac{GHGINT_{\text{local}}}{GHGINT_{\text{mean}}} \tag{6.4}$$

where $GHGINT_{\text{local}}$ represents the GHGINT of the region in which the data center is located. $GHGINT_{\text{mean}}$ represents the average value for the GHGINT of all regions that support that Geo-DisC system. It is worth noting that $\frac{GHGINT_{\text{local}}}{GHGINT_{\text{mean}}}$ is greater than one if the $GHGINT_{\text{local}} > GHGINT_{\text{mean}}$ that makes the C_{eq} greater than the actual carbon emission of that data center. The "equivalent carbon emission" is needed to be separately calculated at the granularity of data centers, and then the results can be aggregated as the "equivalent carbon emission" of the whole system. Figures 6.26 report the carbon emission and "equivalent carbon emission" of a Geo-DisC system under any of these virtual taxes compared with no virtual tax scenarios.

6.2.6 Summary

In this section, the new proposed algorithm was compared with other algorithms under different scenarios. The results show that the new algorithm can minimize the carbon footprint when it is maximizing the profit with considering the trade-off between profit and carbon. A higher carbon tax can significantly reduce the carbon footprint, but it also reduces the profit of the system. In the states and provinces which the carbon tax is not in place introduction of a virtual carbon tax can force the algorithm to reduce the carbon footprint while these virtual carbon taxes will be added to the profit at the end.



(a) GHGINT carbon emission

(b) GHGINT "equivalent carbon emission"

(c) GHGpCapita carbon emission

(d) GHGpCapita "equivalent carbon emission"

(e) MGHGINT carbon emission

(f) MGHGINT "equivalent carbon emission"

Figure 6.26 Geo-DisC under CPAS algorithm with different virtual "GHG indicator" taxes

6.3 Server Power Metering Validation

In the previous sections, an improved model for energy and carbon metering of a distributed cloud have been presented. In this section, first, the proposed model in Equation (3.7), which is the main formula for all the power measurements, is validated on real servers. Then, the VMs live migration power consumption, Equation (3.6), is validated based on the real data.

It is worth noting that since energy consumption of the whole distributed cloud is the summation of energy consumption of individual servers and other equipments (as defined in Equation (3.5), by validating the building blocks of that formulation, the whole formula will be validated.

As mentioned in the related work section, in most of existing power prediction models it is shown that energy consumption has a linear relation with resource usage. In our model, we try to consider as many as possible parameters that may have an effect on energy consumption. Therefore, we expect that our proposed piecewise-linear model works in the same manner with a higher accuracy.

6.3.1 Experimental Setup

In order to evaluate the proposed energy model on a real server, following steps are taken. To calibrate each server, first, a set of different stress tests are run on that specific server, and for each test execution, PMC counters, resource utilization, and energy consumption of the server are measured and logged in a dataset table. Then, models in Kansal *et al.* (2010), Bertran *et al.* (2010b), Farrahi Moghaddam *et al.* (2012b), and our piecewise-linear model are built using linear and piecewise-linear regression techniques. To determine the prediction error rate of each model, a 10-fold cross validation is used; the dataset is divided in 10 random subsets, and the model is trained (regression techniques) using 90% of the dataset (9 subsets) and is tested on the remaining 10% (the other subset). This process is repeated for all the subsets in a total of 100 times. The final error rate is the average of all 100 validation errors.

A custom made stress test application is written in C++ language in order to load different parts of a server with different degree of utilization. This stress process is able to simultaneously load the CPU (arithmetic and/or memory instructions), network, and hard disk with different loads by using multiple threads. A total number of 512 stress test with random generated degree of load in CPU, memory, network, and disk is run on each server.

For collecting the PMCs, we used the Linux kernel profiling tool (“perf”) to collect the task clock msecs, context switches, CPU migrations, page faults, CPU cycles, instructions, cache references, cache misses, L1 dcache loads, L1 dcache stores, L1 dcache prefetches, L1 icache loads, L1 icache prefetches, Last Level Cache (LLC) loads, LLC stores, and LLC prefetch counters. For collecting the network, disk, memory, and CPU use information, we used the

Linux performance analysis tool (“dstat”) to measure the amount of disk read/write activity (MB/s) and the amount of network send/receive activity (MB/s).

6.3.1.1 Server Power Metering Setup

For the tests, GSN servers (Dell PowerEdge R710, 16x Intel Xeon E5530 @2.40GHz, 48GB RAM) are selected to perform energy consumption analysis. The energy consumption of the servers is measured by a Raritan PDU device (PX DPXR8A-20L6) and a backup power source (ServerTech Sentry Switched Cabinet power Distribution Unit (CDU) Version 6.0h). These two servers are connected through gigabit Ethernets.

Each test is carried out for around 100 seconds, and, during the test, the measured power consumption of the server is recorded by the PDU and CDU.

6.3.1.2 VM migration Power Metering Setup

For the VM migration energy consumption test, a VM is created on the source server using KVM hypervisor and then a process is run on that VM (in our case, we used a video streaming process). During the whole test, the process will be active and will be streaming to a client. While the migration process is initiated and is in process, both servers energy consumption will be measured in addition to the PMC counters and resource utilization. Theses recorded values plus the validated model will be used in the following section in order to validate the migration energy consumption model.

6.3.2 Server Power Metering Validation Results

Matlab linear regression and the ARESLab piecewise-linear regression is used to build the Kansal *et al.* (2010), Bertran *et al.* (2010b), and Farrahi Moghaddam *et al.* (2012b) models and our model from collected data[5] on two GSN servers under different loads, and the following

[5]The data is publicly available to community for research purposes in http://www.synchromedia.ca/cadcloud

regressions emerge:

$$p^{(s)}_{Kan}(t) = 225.41 + 1.47d + 4.85n + 75.88c + 49.02m \tag{6.5}$$

$$p^{(s)}_{Ber}(t) = 259.02 + 16.12c_m + 21.04ch_m + 2.38br + 43.98L1_{dl} + 56.35m_b \tag{6.6}$$

$$\begin{aligned} p^{(s)}_{Far}(t) = {} & 185.33 + 2.85d - 0.76n + 6.85c + 21.58c_m + 1.47ch_m + 51.15br \\ & +65.13L1_{dl} + 77.05m_b \end{aligned} \tag{6.7}$$

$$\begin{aligned} p^{(s)}_{CADCloud}(t) = {} & 349 + 458\lfloor c - 0.154\rfloor - 651\lfloor 0.154 - c\rfloor - 367\lfloor m_b - 0.967\rfloor \\ & -64.8\lfloor 0.967 - m_b\rfloor - 946\lfloor 0.0032 - c_m\rfloor - 45.2\lfloor 0.32 - c\rfloor \\ & -83.4\lfloor 0.0656 - n\rfloor - 408\lfloor c - 0.0425\rfloor - 340\lfloor m_b - 0.929\rfloor \\ & -25.2\lfloor L1_{dl} - 0.771\rfloor - 6.85\lfloor 0.771 - L1_{dl}\rfloor + 253\lfloor m_b - 0.886\rfloor \end{aligned} \tag{6.8}$$

where $\lfloor\ \rfloor$ denotes that the enclosed quantity is equal to itself when its value is positive, and zero otherwise. The parameters d, n, c, m, c_m, ch_m, br, $L1_{dl}$, and m_b represent normalized disk, network, CPUs, memory, CPU-migrations, cache-misses, branches, L1-dcache-loads, and memory-buff utilization, respectively. All other parameters with high correlation with one of these parameters are not participating in the models.

Based on 10-fold cross validation, our piecewise-linear model can predict the power consumption of these particular servers with a 2.1% error (std=.018%). In comparison, the error rate of the models proposed in Kansal *et al.* (2010), Bertran *et al.* (2010b), and Farrahi Moghaddam *et al.* (2012b) are 4.2%, 3.8%, and 3.5% with standard deviation of .006%, .010%, and .192%, respectively.

By customizing the degree of complexity of piecewise-linear model simpler and more complex model can be achieved based on application needs. In Figure 6.27, the simplest power prediction model is illustrated based on CPU usage:

$$p^{(s)} = 296.8 + 3.36\lfloor c - 2.42\rfloor - 25.01\lfloor 2.42 - c\rfloor$$

where $p^{(s)}$ represent the power consumption of the server, c represents the number of utilized CPU (not normalized), and circles represent the collected data under different loads. Circles colour represent the I/O relative load of the server (Green: network and Red: disk). Power (VoltAmp) represent the average energy consumption of the server within 100 second test period. The colour code shows that green samples are slightly above red samples which indicates the higher power consumption of network over disk.

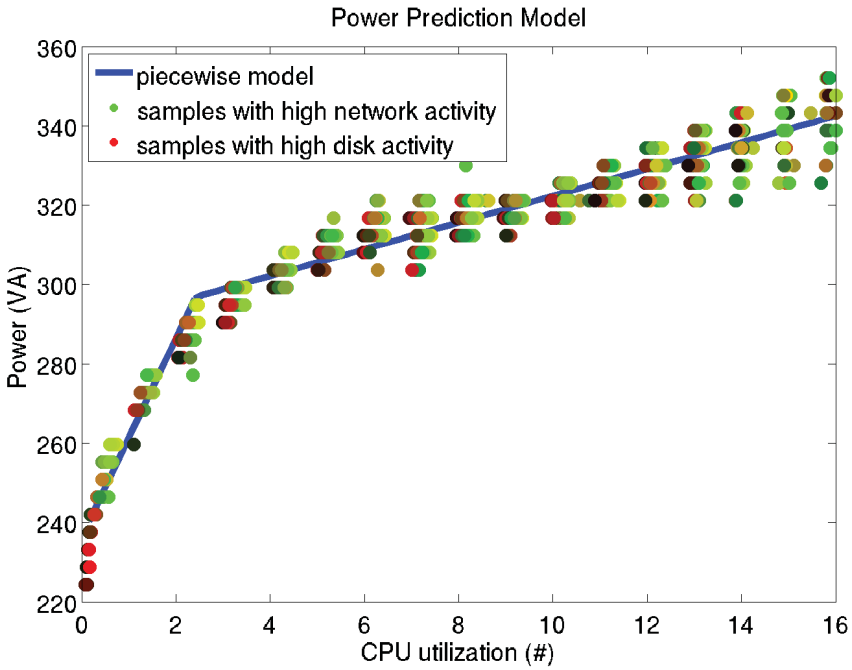


Figure 6.27 Power prediction model

6.3.3 VM Migration Power Metering Validation Results

To validate the VM migration power metering equation, a set of migrations from server A to server B is performed, and the PMCs and resource utilization of the two servers are recorded. By subtracting these measures from measures from servers without any migrations, Δpmc_i, Δc, Δd, Δn, and Δm are calculated. These values can be used to calculate Δp. Then, Equation (3.6) is calculated based on Δps. In Figure 6.28, the actual power consumption readings are compared with predicted values during VM migration. "Server A (Actual)" and "Server

B (Actual)" are the actual readings of server A and server B; "Server A+B (Actual)" represents the summation of the actual power readings; "A without mig (Model)", "B without mig (Model)", and "Migration (Model)" are the predicted server A and server B power consumption, and the migration power consumption respectively; and "A+B+Mig (Model)" represents the summation of all the predicted power values. As can be seen from the figure, the model can predict the migration power consumption within the accuracy our model.

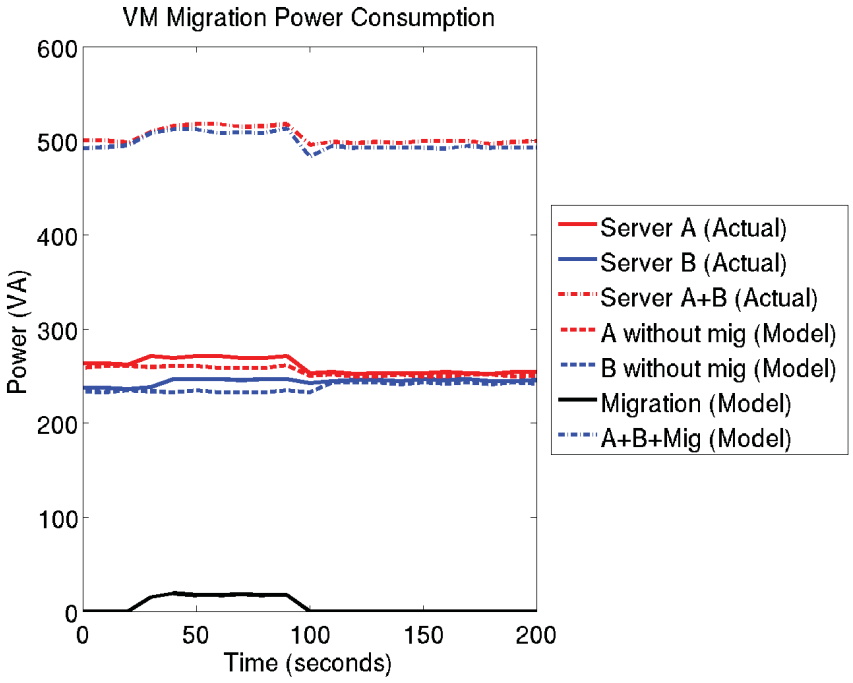


Figure 6.28 VM migration power prediction

6.4 Low-Carbon Web Application Load Balancing

In order to examine the performance of new algorithm on energy efficiency and carbon footprint reduction, we test it on a Matlab simulation platform[6]. A Distributed Cloud (DisC) (Van der Merwe *et al.*, 2010; Wood *et al.*, 2010; Farrahi Moghaddam *et al.*, 2011) is simulated and tested under state-of-the-art methodologies and the proposed MLGGA, namely Carbon-Aware Distributed Cloud (CADCloud). Different conditions are considered to test the pro-

[6]http://www.greenservices.info/2011/10/simulation-environment.html

posed algorithm. In the following, first, the experimental setup is explained, and next, the performance study and energy diversity study of the new algorithm are presented in large scale in a fictitious network to test the algorithm under certain conditions where only some type of energy sources such as solar are considered rather than grid mix. Last, the algorithm is tested in a simulation environment which uses real data from energy mixes to weather temperatures and carbon regulations.

6.4.1 Experimental Setup

The simulation platform is set up in the Matlab environment, which is suitable for experimental simulations, including optimization. On this platform, data centers can be created in geographically distributed locations around the globe.

These data centers are reliably connected by high speed links, which enables seamless VM migration[7] among them. The feasibility of the proposed network structure is based on real VM migration tests performed in the GSN[8] project and in CloudNet Wood *et al.* (2010).

At each data center, the power consumption for utilities, servers, VM migration, and on/off status changes for servers and data centers is simulated. The server specifications used in the simulation experiments are selected so that they are identical to the server specifications used in GSN project servers used for model validation in section 6.3.

In order to correlate the simulation and real server results, performance monitoring counters, disk and network utilization rate, and renewable power generation data are recorded on real systems (GreenStar Network project), and these data are played back during data generation for the simulation experiments.

Several servers can be considered at each data center, which is connected to a primary renewable source of energy with a "greenness factor" g. This source of energy can be intermittent, such as solar or wind, or permanent, such as hydroelectric or geothermal.

[7]Please refer to Equation (3.5) and Equation (3.6) for more information on carbon footprint calculation.
[8]http://greenstarnetwork.com

In each simulation, in order to put the cloud under stress, a set of VMs is created and launched. All the data centers are connected to a secondary and permanent non clean source of energy, which will be used when the primary source of energy is not available. At each data center, a battery bank is considered to store unused energy at peak clean energy production. This stored energy is for use when not enough clean energy is being produced. In each server, the CPU, memory, network, and storage usage of the operating system and VMs are simulated. The batteries are empty at the beginning of each experiment, and they charge during the simulation.

There are 60 data centers containing 3000 physical servers in this evaluation test, located in 24 cities, around the globe. Each data center is powered by two randomly selected primary and secondary sources of energy.

Snapshots of simulation environment for cities are illustrated in Figure 6.29. As it is depicted, each data center is illustrated with a red or green filled circle. Red circle means that data center is using a source of energy with a g factor less than 0.5, and green circle means that data center is using a source of energy with g factor greater than 0.5. The type of source of energy for each data center is illustrated as an icon in the middle of the circle. Available source of energies in this simulation are solar, wind, hydro, nuclear, and grid (natural gas-based). As it is shown, hydro and nuclear source of energies are always green, and grid source of energy is always red. For solar and wind source of energies, it depends on existence of sun and wind, and also on the amount of energy stored in the batteries.

6.4.1.1 Optimization Problem

To evaluate the efficiency of the proposed algorithm, the MLGGA algorithm is compared with the GGA and FFD algorithm which are used in other works for energy efficiency in virtualized data center environments. Carbon footprint and energy consumption of the network are also measured when there is no optimization (NO-CONS) in order to have a baseline in the comparison of the results of these algorithms. This baseline show how much energy and carbon is saved in each algorithm. As shown in the literature review section, there are things which can be done to improve the result of the GGA in energy efficiency in virtualized data center

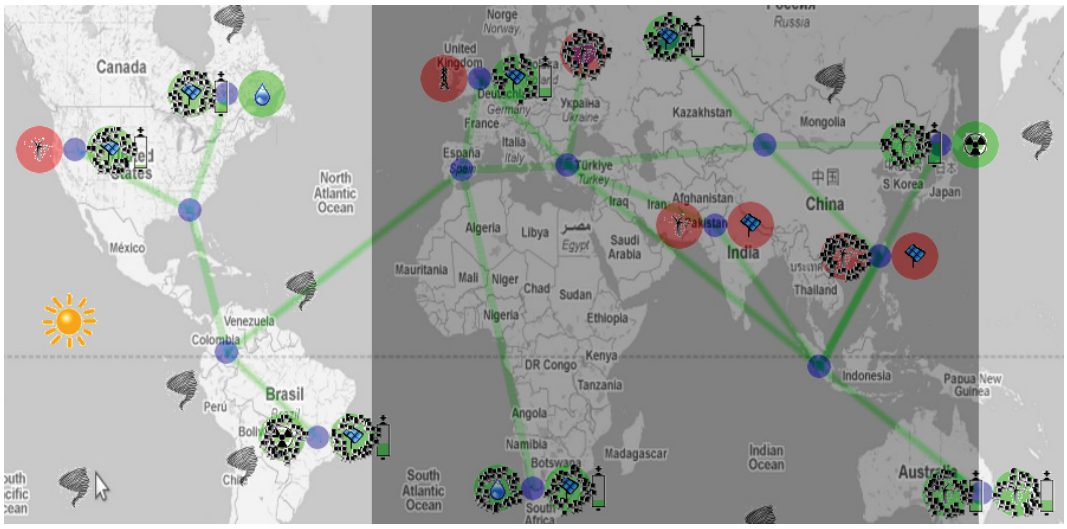


Figure 6.29 Distributed cloud in 11 cities.

environment. Here, we use the same improvements for both GGA and MLGGA as described in previous works. The only differences between the two algorithms implementation are the crossover and mutation operators, and the rest of the algorithms are exactly the same, and both algorithms benefit from the enhancements.

For MLGGA, two algorithm is developed: carbon-aware MLGGA (MLGGA-CA) and energy-aware MLGGA (MLGGA-EA). MLGGA-CA is used to optimized the carbon footprint of the DisC when carbon reduction is the objective, and MLGGA-EA is used when energy efficiency is the objective of the problem. For these two algorithms, the only difference is in the cost function.

For GA-family algorithms, the chromosome is formed from integer numbers, which each number indicate a server. The number of genes is equal to number of VMs. In a table, each server is assigned to a data center, and in another table each data center is assigned to a city and is assigned to a source of energy as its primary source of energy. Therefore each chromosome defines the state of VMs and the source of energy which they are powered with.

For all the algorithms which are compared in this experimental setup bin packing conditions explained in Equation (1.1) are respected.

6.4.2 MLGGA Performance Analysis on Large Scale CADCloud

In this section, the energy consumption model validated in the previous section is used in a simulation environment. The goal is to show that advanced multi-level bin packing optimizers are more efficient for the CADCloud type of problem. Below, we compare the results of the various server consolidation techniques on a CADCloud.

To evaluate the efficiency of the MLGGA, we compare it with the traditional server consolidation techniques applied at data centers. Specifically, in a typical scenario, the results of carbon footprint reduction using the MLGGA and the GGA, the First Fit Decreasing algorithm, and Swarm-based server consolidation are compared with the No Consolidation (NO-CONS) option. In the NO-CONS situation, there is no energy efficiency or carbon footprint reduction optimizer running.

6.4.2.1 MLGGA Comparison Results

Simulation was carried out for 168 hours (7 days) for each method, and total carbon footprint and energy consumption were recorded.

As shown in Figure 6.30 and Figure 6.31, carbon is significantly reduced by the MLGGA-CA in the CADCloud simulation (The offset value located in top of the figures need to be added to figure values in order to achieve the actual energy and carbon values). For the energy consumption, the MLGGA-EA is slightly better than the MLGGA-CA.

As shown in Table 6.14, in each set of loads, the G factor for the energy aware methods are in the same range, and the G[9] factor of the MLGGA-CA is higher than that of all the energy aware consolidation methods. With an increase in load, this advantage will decrease from 27.88 to 17.32 when the load increases from 25% to 75%. The carbon footprint is significantly lower with the MLGGA-CA than it is in the others, and the MLGGA-EA ranks second. The carbon footprint of GGA, Swarm, and FFD are in the same range. The carbon footprint increases with an increase in load. Energy consumption is lower with the MLGGA-EA relative to that of

[9]Refer to Section Section 3.1.3 for more information

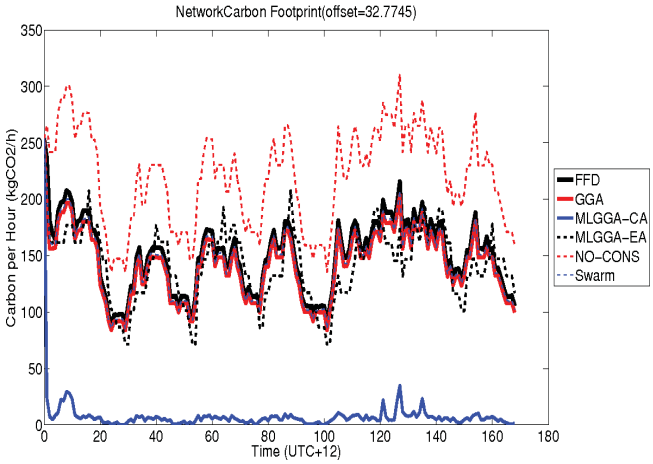


Figure 6.30 Comparison of various methods with respect to carbon footprint. (50% load)

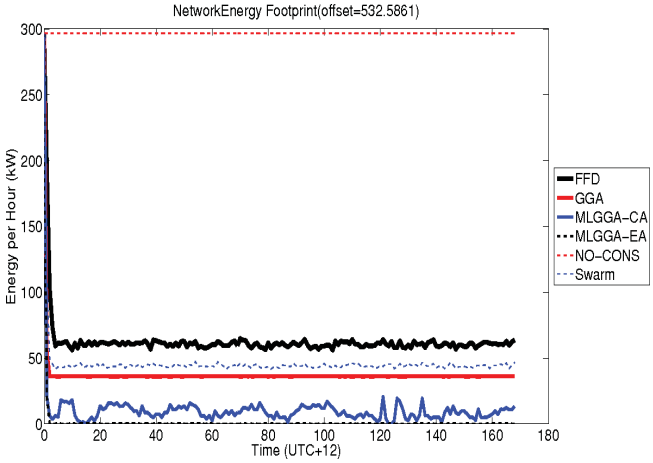


Figure 6.31 Comparison of various methods with respect to energy consumption. (50% load)

the others, followed by the MLGGA-CA. The energy consumption for the GGA, Swarm, and FFD are in the same range. The energy consumption increases with an increase in load for all algorithms.

According to Table 6.14, the MLGGAs are less sensitive to sun and wind, since they are able to switch from solar and wind power to another source of renewable energy. This sensitivity will increase with an increase in load for the MLGGA, since with a higher load there are fewer

options for switching VMs to low carbon servers. The carbon footprint of the cloud with respect to weather conditions is presented in Table 6.15.

Table 6.14 MLGGA performance study: 24-hour carbon and energy measurements

Load	Algorithm	Carbon (kg/-day)	Energy (kWh/-day)	G (%)	SunSensitivity ($kgCO_2$ /day)	WindSensitivity ($kgCO_2$ /day)
25%	MLGGA-CA	**347.7**	6243	**93.812**	**244.6**	**82.2**
25%	MLGGA-EA	1881	**6134**	65.929	798.6	1605.7
25%	GGA	2403.3	7430	64.059	1160.1	1100.8
25%	Swarm	2405.3	7434.8	64.054	1201.9	1094.2
25%	FFD	2415.7	7468.3	64.061	1175.2	1128
25%	NO-CONS	5501	17006.7	64.059	2655.2	2519.7
50%	MLGGA-CA	**895.1**	12968.3	**92.331**	**1402.6**	1707
50%	MLGGA-EA	4060.8	**12743.3**	64.592	2264.8	**1557.6**
50%	GGA	4068.8	13607.3	66.775	2142.7	2726.8
50%	Swarm	4076.2	13628.3	66.676	2151.2	2743.8
50%	FFD	4098.2	13719	66.808	2137	2769.2
50%	NO-CONS	5952.7	19906.7	66.775	3134.7	3989.2
75%	MLGGA-CA	**3412**	19105	**80.157**	3713.5	3150.8
75%	MLGGA-EA	6602.7	**18876.7**	61.137	3145.7	**2409.5**
75%	GGA	6459.3	19310	62.832	3031.5	2627
75%	Swarm	6474.2	19346.7	62.819	**3009.7**	2568
75%	FFD	6508.5	19465	62.849	3087.2	2564
75%	NO-CONS	7555.2	22585	62.832	3545.8	3072.7

6.4.3 Energy Diversity Study

In this section, the results of the carbon footprint optimization of a CADCloud are compared, considering diverse sources of energy. To build a complete set of experiments on the CADCloud, we considered several scenarios. Different combinations of energy type were used with the same number of data centers powered by each type of energy.

We compare the carbon footprint reduction performance of the CADCloud in these scenarios. Also, the sensitivity of each scenario to intermittent sources of energy is measured by a sensitivity factor.

To measure the sensitivity of each scenario to the percentage of power usage of data centers, each scenario was tested under various loads. To obtain a comprehensive view of the load sensitivity of the various scenarios, we selected a range of loads, from 10% to 90%.

Table 6.15 MLGGA performance study: 24-hour carbon measurement with weather change

Load	Algorithm	Sun(100%) Wind(100%)	Sun(100%) Wind(67%)	Sun(100%) Wind(33%)	Sun(50%) Wind(100%)	Sun(50%) Wind(67%)	Sun(50%) Wind(33%)
25%	MLGGA-CA	347.69	384.59	400.96	469.07	506.21	525.4
25%	MLGGA-EA	1881	2665.3	2942.5	2278.5	3050.2	3357.9
25%	GGA	2403.4	2889.7	3122.5	2974	3458.9	3722.8
25%	Swarm	2405.3	2893.8	3126.9	2997.7	3496	3735
25%	FFD	2415.6	2914.8	3140.7	2976	3502.6	3755.2
25%	NO-CONS	5501	6614	7146.8	6807	7916.7	8520.9
50%	MLGGA-CA	895.14	1279.1	1759.2	1271.5	2082.3	2683.5
50%	MLGGA-EA	4060.9	4355.8	5098	5281.7	5309	6321.4
50%	GGA	4068.9	5143.7	5877.2	5141.1	6194.2	6968.5
50%	Swarm	4076.2	5155.1	5887.7	5138.6	6221.5	6985.5
50%	FFD	4098.2	5194.5	5933.8	5161.6	6252.3	7018.1
50%	NO-CONS	5952.7	7525.2	8598.3	7521.4	9062	10195
75%	MLGGA-CA	3412	4745.7	5282.6	5025	6630.2	7355.4
75%	MLGGA-EA	6602.7	8028.2	8291.1	8265.4	9585.5	9789.7
75%	GGA	6459.3	7751.9	8195.2	7990.5	9205.8	9757.3
75%	Swarm	6474.2	7777.8	8210.2	8007.6	9273.6	9695.6
75%	FFD	6508.5	7820.2	8261.8	8120.1	9315.8	9785.3
75%	NO-CONS	7555.1	9067	9585.6	9346.2	10768	11413

The other sensitivity measure of the CADCloud is the sensitivity of intermittent sources of energy to weather conditions. To cover all the important parameters of the CADCloud in all the scenarios, the simulator was run several times on the simulation platform. These results are reported in section 6.4.3.1. Each scenario was run for 168 hours (7 days) in a simulation environment under different weather and load conditions.

As mentioned previously, different sources of energy are used in different scenarios. The percentage of use of each source of energy in each scenario is listed in Table 6.16.

6.4.3.1 Results

The carbon footprint was measured as shown in Figure 6.32 for the set of scenarios.

As expected for the scenarios, the carbon footprint is larger for higher loads. The carbon footprint is lowest in Scenario 3, followed by Scenarios 8, 6, and 11, and then by Scenarios 14, 13, 15, and 2. The highest carbon footprint was measured in Scenario 4. The scenarios with

Table 6.16 Energy diversity study: ration of sources of energy in different scenarios

Scenario	Solar	Wind	Perm. Green	Gas
1	100%	0%	0%	0%
2	0%	100%	0%	0%
3	0%	0%	100%	0%
4	0%	0%	0%	100%
5	50%	50%	0%	0%
6	50%	0%	50%	0%
7	50%	0%	0%	50%
8	0%	50%	50%	0%
9	0%	50%	0%	50%
10	0%	0%	50%	50%
11	33.33%	33.33%	33.33%	0%
12	33.33%	33.33%	0%	33.33%
13	33.33%	0%	33.33%	33.33%
14	0%	33.33%	33.33%	33.33%
15	25%	25%	25%	25%

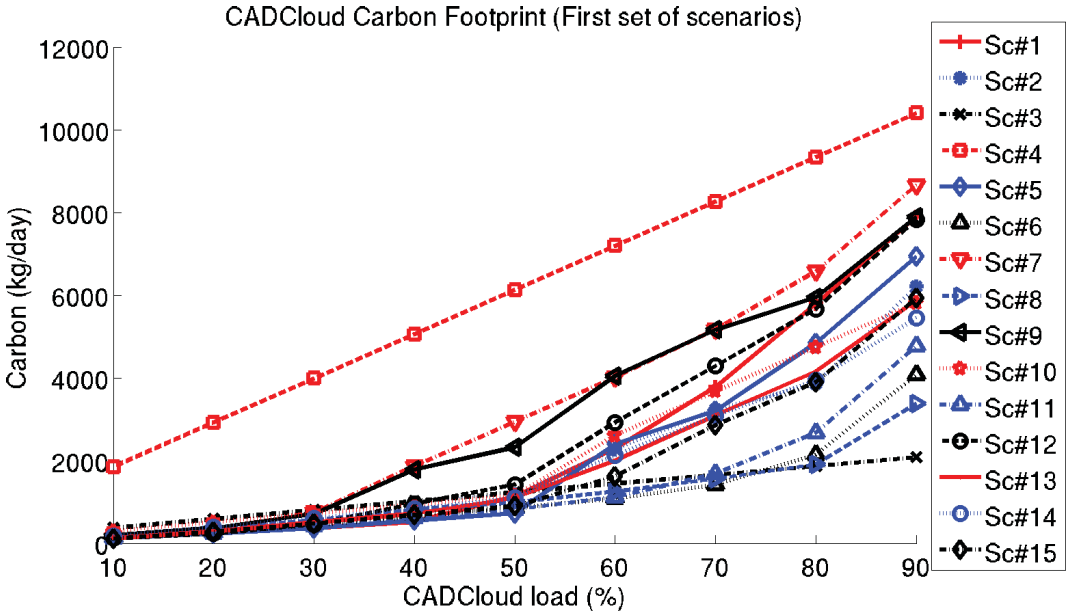


Figure 6.32 CADCloud Carbon measurement

permanent clean energies like (hydroelectric power) are at the cleanest, followed by scenarios with intermittent clean energies (solar and wind), and permanent semi clean energies (gas).

For the sensitivity calculations, the scenarios were tested under cloudy conditions (50% of solar power plant capacity) and less windy conditions (33% and 66% of wind power plant capacity)

to obtain the sensitivity of the scenarios to solar and wind energy sources. The sensitivity of the scenarios to solar and wind are depicted in Figures 6.33 and 6.34.

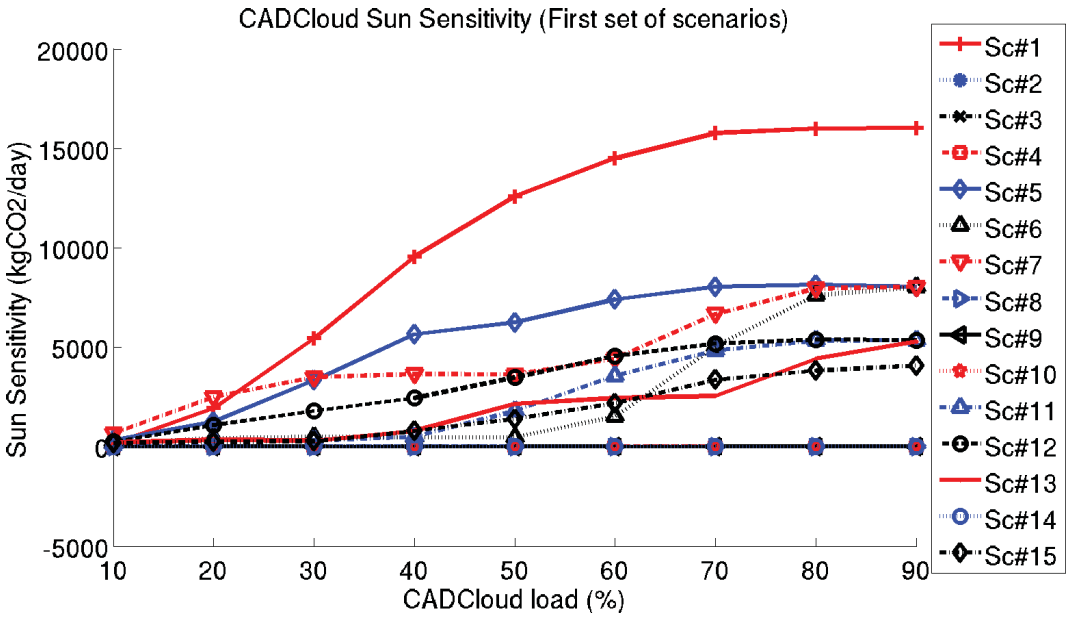


Figure 6.33 CADCloud Sun sensitivity

As expected, the scenarios with mostly solar power are the most sensitive to sun (Scenario 1 100% solar). The least sensitive scenarios are those using the most permanent clean and permanent semi clean energies.

As another CADCloud measure, the G factors of scenarios are depicted in Figures 6.35. It is worth noting that any two of the following parameters are sufficient to represent the carbon reduction performance of a CADCloud: energy consumption, carbon footprint, and greenness factor. As shown in Figure 6.35, the greenness factors of Scenarios 3 and 4 are constant. This is because no intermittent source of energy was considered in these scenarios. The greenness factor for each energy type was taken from the Current State of Development of Electricity-Generating Technologies table in Lenzen (2010).

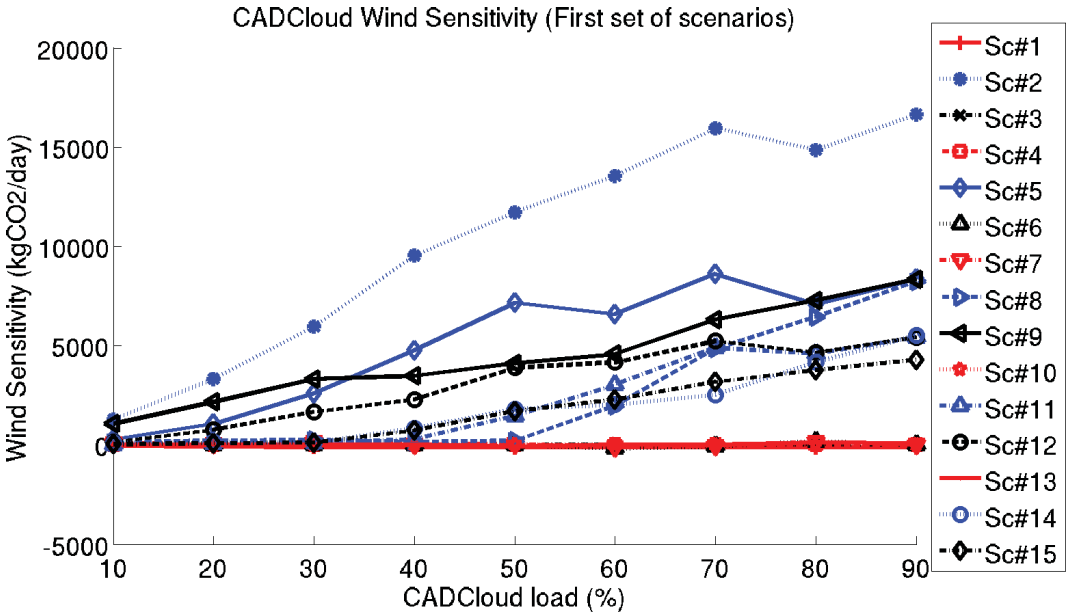


Figure 6.34 CADCloud wind sensitivity

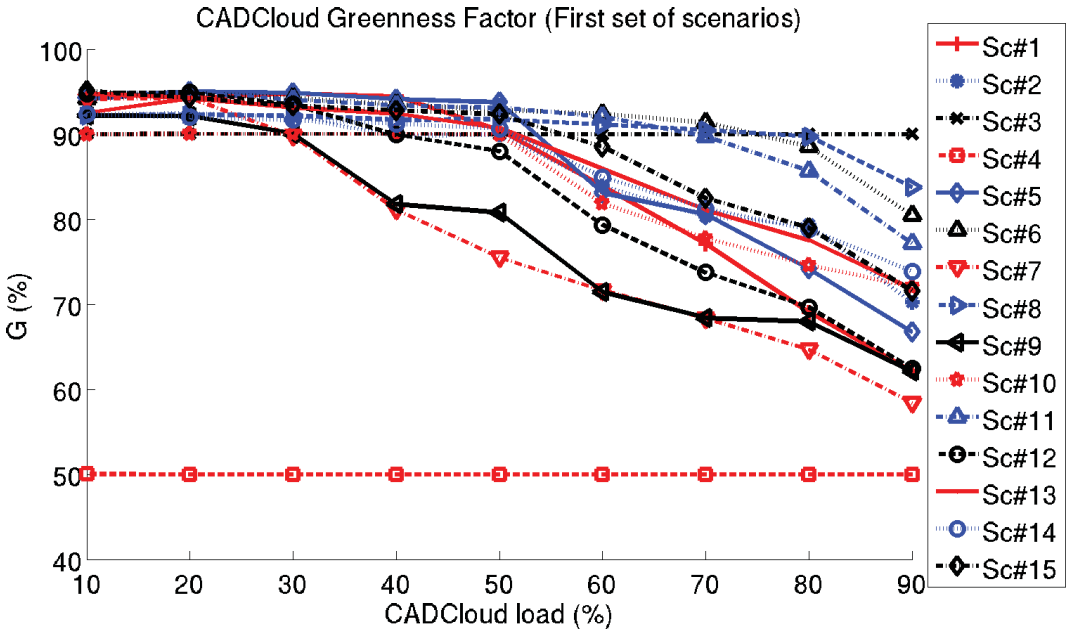
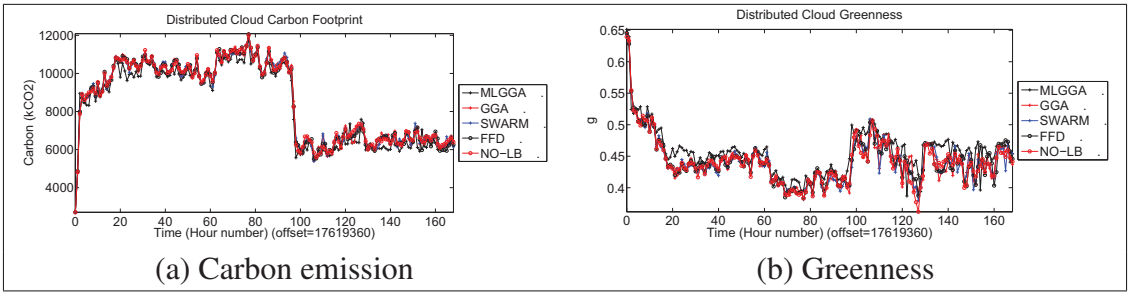


Figure 6.35 The G factors of various scenarios

6.4.4 MLGGA Performance Study on Real Data

In previous sections, the performance of MLGGA algorithm was compared with other algorithms in a simulated environment which the scenarios are based on future promises. For example in today total energy production, solar energy play a very small role, but in future this role will be changed significantly. Therefore, the scenarios with lots of solar energy as their power source maybe more realistic in future cases. To have a realistic analysis on MLGGA algorithm, we examine it under a simulation platform working with real data similar to what we used in HPC experimental setup section. The differences here are that instead of HPC jobs, web applications are deployed here which they have less processing requirements and they run for a longer period of time in compare with HPC jobs. The associated results are presented in Figure 6.36.



(a) Carbon emission (b) Greenness

Figure 6.36 Geo-DisC under load balancing algorithms

As it can be seen in the Figure 6.36-a the carbon footprint of the system is decreased with use of MLGGA algorithm, and in the Figure 6.36-b, the greenness of the whole system is improved.

6.4.5 MLGGA Convergence Time

In our implementation in Matlab, in average, MLGGA takes about 45 minutes to complete maximum one million iterations, and in average, it can reach to a convergence point in one third of this max iterations (15 minutes). However, with introduction of a bias population pool to the algorithm this time can be significantly reduced. The bias population pool can be achieved by using one of greedy algorithms.

CONCLUSION

This thesis focuses on network of data centers (NDCs) with cloud capabilities, and models their operation and environmental impacts toward proposing design and management solutions in order to reduce NDCs' negative footprint while keeping their profitability in a reasonable range. The study includes, first, designing a new management and control system, and then comparing its performance with that of the state-of-the-art methodologies. The main objective of the proposed system is to optimize the profit and carbon footprint of a distributed cloud together while considering the trade-off between them. In this research, two types of loads, HPC and web applications, were considered for the distributed clouds, and for each type of application, specific solutions were provided based on their specific characteristics and requirements.

In chapter 2, first, a baseline system was introduced based on existing approaches in the state-of-the-art literature. This baseline includes models and methods for a network of data centers to calculate the energy consumption, carbon footprint and profit of the system and ultimately optimize these metrics based on particular goals such as maximum profit and minimum carbon footprint. Then, in Section 2.2, the description of a new system was presented to improve the baseline system in achieving its goals. This new system includes new metrics, new models, and new methods such as new cooling model, new heuristic algorithm for load balancing, new HPC scheduler, new server power metering model, and new metrics for HPC scheduler. In addition, the design of the new system shows the connectivity between several active modules working as managers and controllers in the system and their relation to each other.

To fulfil the modeling objective of this thesis, first, a model is introduced in Section 3.1.1 to calculate the profit of an NDC in a CPU core during an hour, namely PpCHG. This metric is used to optimize the profit of the NDC. A 2D illustration of PpCHG in an NDC shows the best position of jobs in the core-time space. This information is then used by the scheduler to optimize the profit of the whole NDC. The experimental results show the effectiveness of the new metric and new methods in various cases. Next, in Section 3.1.2, the power metering models of servers are improved. The model has been calibrated and validated using measured data from experiments on two servers from the GSN network at the server and also migration

levels. The model coefficients have been obtained using a piecewise-linear regression on the measured data. According to the validation results, the proposed model is able to predict the energy consumption and carbon footprint of a CADCloud with accuracy of higher than 97%. Then, a model for cooling system of data centers is presented which is a complete and generic model for the cooling system (chiller-based with cooling tower). This model help with having an accurate calculation on the energy consumption of the support system as well as optimizing this energy consumption. This model also helps with calculation of PUE of the system.

In chapter 4, a new scheduler is introduced which works based on DVFS. This scheduler works based on PpCHG metric which was mentioned above. The scheduler will optimize the frequency of the CPU cores in order to maximize the PpCHG metric. Then, it will use the optimal frequency to schedule the new jobs to the servers. In addition, a new metric is introduced acting as an intermediate force for carbon footprint reduction, namely virtual carbon tax. This virtual carbon tax acts as a carbon tax during the action of scheduler to force it to reduce the carbon footprint in fear of higher costs and consequently lower profit. However, at the end, the virtual carbon tax is part of the profit of the system and will be aggregated to that.

In addition, in chapter 5, a new heuristic algorithm is introduced for use in distributed clouds. This algorithm which is an extension of GGA algorithm can be used with different objectives such as energy efficiency and carbon footprint reduction (MLGGA-EA and MLGGA-CA). A simulated environment in Matlab is used to prepare a test platform for comparison of the new algorithm with the state-of-the-art methodologies. Real data collected from real distributed clouds is used to emulate the platform measurements. Comparison of results shows that MLGGA-CA is an over all better solution for distributed clouds. It is tested under different level of loads, and in all cases MLGGA-CA was able to provide an optimum solution for carbon footprint reduction problem. However, for energy efficiency, the MLGGA-EA has slightly better results. Therefore, usage of MLGGA-EA is not recommended due to its much higher carbon footprint. In addition, it has been verified that in a CADCloud, energy efficiency and carbon footprint reduction are not necessary correlated.

To evaluate the proposed models and methods, a simulation platform is designed and implemented. In this platform, computing components as well as support components, energy sources, and weather are simulated. For accuracy in many areas, real data is used to model the behaviour of the components. In addition, a batch processing is implemented in the simulation platform for scenarios with a high number of cases. Also, a caching mechanism is considered in the system to speed up the process of simulations.

For HPC applications, the performance study shows that the new scheduler has a better profit performance than state-of-the-art schedulers. However, its carbon footprint may not be minimum among all the algorithms. In virtual carbon tax study, it was observed that with the help of VCT this higher performance can be tuned towards the higher carbon footprint reduction accordingly. The scheduler was also tested under different seasonal data, and it was shown that different seasons can significantly change the results of the scheduler under the same load of jobs. In another study, the importance of accurate cooling system modeling was tested based on real weather information, and it was observed that without accurate cooling system modeling, the error margin for the results is high.

According to the simulation results, the performance of MLGGA-CA reduces with an increase in load of the DisC, because of lower green options in the DisC for algorithms to move the VMs. In addition, an energy diversity study has been also performed using the simulation environment, and it has been observed that scenarios that include permanently-available green energy sources are the most robust and valuable scenarios. These scenarios have both a smaller carbon footprint and a lower sensitivity to changes in the weather conditions. However, permanent green energy sources are not available everywhere. The second choice for low carbon footprint sources of energy is intermittently-available green energy. However, because of their high sensitivity to weather conditions, scenarios with intermittent green energy sources alone are not recommended. A scenario with a proportion of intermittently-available green energy sources and permanently-available semi-clean energy sources is a less green, but it is a more reliable compromised choice. Overall, it is observed that the best practice scenario and approach for real world application is using clean and renewable sources of energy in combination of the

MLGGA-CA method for VM management. Using only clean but intermittent sources of energy is not recommended without the availability of clean permanent sources. If clean sources of energy are limited, using semi-clean sources of energy, such as natural gas, in combination with clean sources is highly recommended.

Here, specific contributions related to objectives of this thesis are listed:

- System design (Obj #1):
  - Defining the data structure of the system.

    As mentioned earlier, in this objective, most of the metrics used in the state-of-the-art researches and new metrics are used in the models. Therefore, a comprehensive data structure is needed to be defined in order to hold the necessary data required by algorithms.
  - Defining the module structure of the system.

    In addition to have the data model, it is important to identify the modules and their means of communication with other modules.
  - Considering the profit-carbon-performance aware strategies all together.

    For satisfying this objective of the research, it is necessary to design and implement a new comprehensive scheduler which is able to deliver the goals of this objective which are considering total profit and total carbon emission based on energy mix data, carbon tax, energy price, weather, etc.
- Modeling (Obj #2):
  - Profit-per-Core-Hour-GHz.

    In this research a new metric is introduced to estimate the profit associated with running a CPU core for an hour with the frequency of f. This metric is then used to optimize the profit of each core-hour unit of the system by choosing the most suitable frequency.

  - Cooling power modeling and optimization.

    The amount of power consumed by the supply system accounts for a big portion of total energy consumption of the system and is comparable with IT energy consumption. The cooling system can be very complicated system to model and optimize. In this research a model for cooling system of a typical data center is presented and optimized.

  - Server power metering.

    In this research a power model for individual servers are presented based on their resource utilization and PMC counters.

  - Introduction of greenness factor.

    It is a normalized version of the emission factor of energy mix, which is also introduced for NDC.

  - Introduction of sensitivity to intermittent source of energies.

    This is a measure to show how much the system will be impacted if there is a sudden change in the weather.

- HPC scheduler (Obj #3):

  - Introduction of Carbon Virtual Tax.

    This is a measure which does not exist in real world, and each business need to consider it based on their own goals. In definition, it is acting exactly like the carbon tax, but the amount of calculated carbon tax considered as profit at the end.

  - Introduction of MGhGint to scheduler.

    This is a study for reduction of GhGint factor instead of net carbon emissions.

  - Introduction of Profit-per-Core-Hour-GHz metric.

    This metric is calculated based on many other metrics which are already used in other researches. But, this metric provide a very accurate information for decision making of position of jobs based on total profit of the system.

- Web applications load balancer (Obj #4):

  - A new genetic algorithm is introduced, namely Multi-Level Grouping Genetic Algorithm.
  - MLGGA video presentation.

    A video representation shows the status of the NDC, status of renewable energies, etc, and how the scheduler works.
  - Developing GSN controller.

    A simple controller was developed to load balance the VMs in GSN project.

- Simulation platform (Obj #5):
  - Implementation of cache.

    The optimization and model calculations could be very time consuming. Since some of them are repeating, a cache can really speed up the process.
  - Simulation job generator.

    Since the algorithms need to be tested under different variation of parameters, in this research a simulation job generator is developed which is able to create number of simulation jobs which they vary in different parameters such as size, load, energy, type, price, etc. And then, the simulation job scheduler can assign the jobs to be executed by the simulator, and collect the final results and create comparative plots.

Below is the list of publications related to the research conducted in this thesis:

- Farrahi Moghaddam, Fereydoun, and Cheriet, Mohamed, "Designing a Carbon-Profit-Aware Scheduler Based on Virtual Carbon Tax in Geo-Distributed Clouds," submitted to Sustainable Computing Informatics and Systems, 2013.
- Farrahi Moghaddam, Fereydoun, Reza, Farrahi Moghaddam, and Cheriet, Mohamed, "Carbon-Aware Distributed Cloud: Multi-Level Grouping Genetic Algorithm," (under revision) Springer Cluster Computing, 2013.

- Farrahi Moghaddam, Fereydoun, Reza, Farrahi Moghaddam, and Cheriet, Mohamed, "Carbon Metering and Effective Tax Cost Modeling for Virtual Machines." 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), IEEE, 2012, Pages 758-763.

- Farrahi Moghaddam, Fereydoun, Reza, Farrahi Moghaddam, and Cheriet, Mohamed, "Multi-level Grouping Genetic Algorithm for Low Carbon Virtual Private Clouds." CLOSER 12, 2012, Pages 315-324.

- Farrahi Moghaddam, Fereydoun, Cheriet, Mohamed, and Nguyen, Kim Khoa, "Low Carbon Virtual Private Clouds." 2011 IEEE International Conference on Cloud Computing (CLOUD), IEEE, 2011, Pages 259-266.

- Farrahi Moghaddam, Fereydoun, and Cheriet, Mohamed, "Decreasing Live Virtual Machine Migration Down-Time Using a Memory Page Selection Based on Memory Change PDF." Networking, Sensing and Control (ICNSC), 2010 International Conference on. IEEE, 2010, Pages 355-359.

- Farrahi Moghaddam, Reza, Farrahi Moghaddam, Fereydoun, and Cheriet, Mohamed, "A modified GHG intensity indicator: Toward a sustainable global economy based on a carbon border tax and emissions trading," Energy Policy, Volume 57, June 2013, Pages 363-380.

ANNEX I

DEFINITIONS

1 HPC Job

HPC jobs are type of applications which mainly consume nodes' CPU compute power, while their memory, network, and disk utilization are minimum. Weather simulation, genetic sequencing, animation rendering, and market modeling are all examples of HPC jobs.

2 "Day Number" and "Hour Number"

When dealing with different time zones, different years, months, and days, to have a continuous measure of time, "day number" is often used as the indicator for time instead of human readable calendar dates. The day number is a number which counts each day from the beginning of Gregorian calendar, where its decimal fraction shows the hour, minute, second, and millisecond of that day.

Hour number is defined exactly with the same concept, but it counts the hours since the beginning of Gregorian calendar instead. The Hour number can be calculated by multiplication of the day number to 24.

3 Energy, Power, Carbon, and Carbon-per-Hour

There is usually a bit of confusion between power and energy concepts. The power is the rate of generation or consumption of energy, and its measurement unit could be kilo-watt (kW) or any other energy-per-unit-time unit. While, the unit of energy is usually kilo-watt-hour (kWh). If the power is variable along the time, the amount of energy is calculated by integrating the area under the power curve. There is a separate amount of carbon footprint associated with any of these metrics. For example, Kilograms-CO_2 is a well-known unit of footprint associated with the amount of energy consumed in an interval of time. On the other hand, Kilograms-CO_2 -per-hour is the unit to measure the rate of carbon footprint emissions associated with the power

consumption at a particular instance of time. These metrics are convertible to each other: The footprint associated to an interval of time (measured in Kilograms-CO_2) can be calculated by integrating the area under the power footprint curve (measured in Kilograms-CO_2-per-hour).

4 Server Consolidation

Server consolidation is a strategy for energy efficiency in data centers. The main idea behind server consolidation is to take advantage of virtualization technology to consolidate more than one under-utilized virtual machine (virtual server) on a physical server. This can be done by live seamless migrations, live migrations, or even offline migrations of VMs. In live seamless migration, there is no or minimum interrupt in the virtual machine operation. On the other hand, there is a down-time in operation of the VM without change in its state in live migration, while the state of VM need to be changed to shutdown or hibernate before migration take place in offline migration. With consolidation strategy, physical servers with no virtual machine hosted on them will be shutdown or put on stand by mode to save energy.

5 DVFS

Some CPUs/cores allow hot modification of their frequency without interrupting their operation. This is referred to dynamic voltage and frequency scaling (DVFS) or dynamic voltage scaling (DVS) in literature. This is of great interest because there is a non-linear (usually near cubic) relation between power consumption of a CPU/core and its working frequency. This feature is used in some strategies for energy efficiency purposes in place or along with consolidation. It is worth noting that a few discrete values of frequency are usually accessible in practice (Gandhi *et al.*, 2009).

6 Metrics for Schedulers

Depending on the type of load and also goals of an HPC compute provider, different metrics may be used. Below, a list of popular metrics is provided:

- Execution Time (ET) (Maheswaran *et al.*, 1999): The amount of time a resource spends to finish a task given that resource does not have any other load when that task is assigned: ET(t_i, r_j), where t_i is the task and r_j is the resource.

- Expected Time to Compute (ETC) (Maheswaran *et al.*, 1999): it is defined as the time required to complete a specific job on a specific resource, which could be modeled as the ratio of the job load (in millions of instructions per second (MIPS) or in Operations per second (Ops)) to the capacity of the resource. In ideal setting, ET and ETC should be the same. However, in practice ET could be higher than ETC.

- Expected Completion Time (ECT) (Maheswaran *et al.*, 1999): The wall-clock time at which a task is finished on a resource (after finishing its previously assigned tasks) assuming that task is assigned to that resource: ECT$(t_i, r_j, \mathrm{T}_{j,i} = t_k)$, where t_k are other tasks assigned to r_j before t_i. Please note that the ECT is not a time period.

- Arrival Time (AT) (Maheswaran *et al.*, 1999): The wall-clock time when a task arrives to the scheduler queue. In other words, the scheduler is not aware of the task t_i before AT(t_i).

- Begin-to-Execute Time (BET) (Maheswaran *et al.*, 1999): The wall-clock time a task begins its execution on an assigned resource. The following relations always hold: ECT$(t_i) \geq$ BET$(t_i) \geq$ AT(t_i) and ECT$(t_i) =$ BET$(t_i) +$ ET(t_i).

- Completion Time (CT) (Maheswaran *et al.*, 1999): The completion time of a task is the ECT of that task on the resource that the task is assigned to it. CT$(t_i) =$ ECT$(t_i, r_{\hat{j}})$, where $r_{\hat{j}}$ is the actual resource that the task t_i is assigned to.

ANNEX II

A MODIFIED GHG INTENSITY INDICATOR: TOWARD A SUSTAINABLE GLOBAL ECONOMY BASED ON A CARBON BORDER TAX AND EMISSIONS TRADING

It will be difficult to gain an agreement of all the actors (countries) on any proposal for climate change management, if universality and fairness are not considered. To address this gap, in this work, a universal measure and indicator of emissions to be applied at the international level is proposed. This indicator is based on a modification of the Greenhouse Gas Intensity (GHGINT) measure. It is hoped that the generality and low administrative cost of this measure, which we call the Modified Greenhouse Gas Intensity measure (MGHGINT) (Farrahi Moghaddam *et al.*, 2013), will eliminate any need to classify nations. Classification of countries, for example in the Kyoto Protocol, has been a source of disagreement and also failure because of shift of GHG emissions to developing countries. The core of the MGHGINT is what we call the IHDI-adjusted Gross Domestic Product (IDHIGDP), based on the Inequality-adjusted Human Development Index (IHDI). The IDHIGDP makes it possible to propose universal measures, such as the MGHGINT. We also propose a carbon border tax applicable at national borders, based on MGHGINT and IDHIGDP. This carbon border tax is supported by a proposed global Emissions Trading System (ETS). The proposed carbon tax is analyzed in a short-term scenario, where it is shown that it can result in a significant reduction in global emissions while keeping the economy growing at a positive rate. In addition to single-year GHG emissions, the MGHGINT is generalized to consider the cumulative GHG emissions over two decades, which had almost the same results of the single-year MGHGINT.

1 The benefits of a universal indicator

As mentioned above, there will be no global agreement without an universal indicator at hand. Actually, the past performance of partial agreements that have went into action, such as that of Kyoto Protocol Accord, suffer from several drawbacks including but not limited to:

a. Deindustrialization

b. High volumes of imported emissions

Although other factors, such as high level of inequality and its consequence of highly low wages, have played a critical role in outsourcing of industry to those countries, strong commitment of European Union to its goals of reducing GHG emissions' of the union by 20% to 30% compared to those of 1990 has implicitly driven many heavy industries, such as steel industry, to moved to China. This could result in higher level of global emissions because of exclusion of these regions from the accord and therefore possible use of old and high-footprint technologies in production. At the same time, a large portion of these products with high-level of carbon content will exported back to the union region, and therefore the overall GHG footprint would be even higher than that of business as usual in absence of Kyoto Protocol.

In addition, a universal indicator would enable and justify unilateral actions even in absence of a global agreement. In other words, even if a global agreement does not reached, individual countries could impose a border adjustment or tax to impose mechanisms that implicitly impact the GHG emissions of other regions. An universal indicator enables this move and homogenize the unilateral effort of different regions toward easier acceptance and justification with respect to the world trade organization (WTO) regulations.

2 Lack of a Universal Indicator

There are two major indicators for the GHG emissions of regions:

a. The GHG Intensity indicator (GHGINT)

b. The GHG emissions per Capita (GHGpCapita)

As can be seen from Figure II-1, these two indicators do not behave homogeneously across the nations, and each one highly penalizes a group of counties, while it praises another group. The scale of this divergent behavior, which roots in the fundamental differences between developed

and developing countries, is so high that none of these two indicators has been accepted by all countries. This can actually be seen as the origin of partial coverage and exclusion of developing countries in agreements such as Kyoto Protocol. Although these indicators may be patched in order to make them more adaptive to a group of countries, the scale of changes in status of every country, and especially the high pace of these changes, would require revising these patches every few year. On the other hand, a universal indicator that covers all the regions in a fair and uniform approach would gain global acceptance and also would help avoiding continuous negotiations and changes in the policies. Here, we propose a universal indicator called Modified GHG Intensity (MGHGINT).

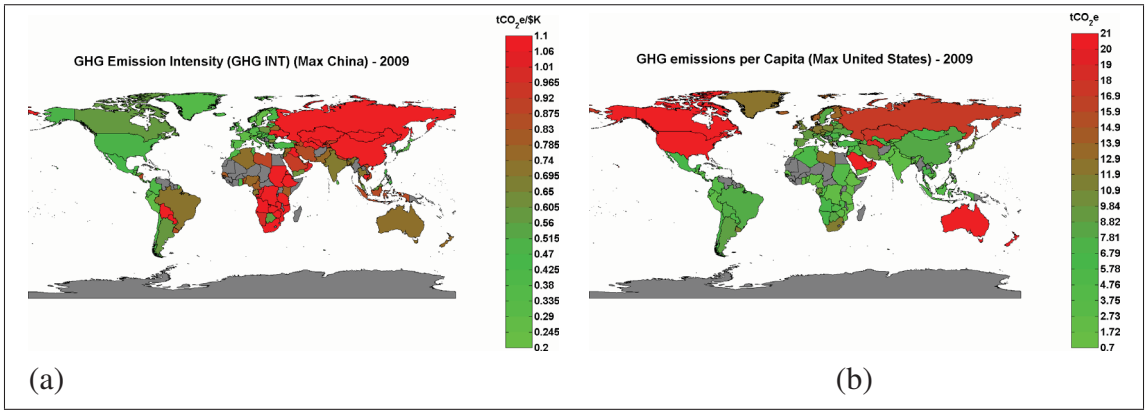


Figure-A II-1 a) GHG Emission intensity (GHGINT) in 2009 (in $GtCO_2e/\$B$). b) GHG emissions per capita in the same year (in $GtCO_2e$/Million Capita). *Data Sources:* US Energy Information Administration, World Bank, United Nations Statistics and Research Database, International Monetary Fund, and United Nations Development Programme.

3 The Potentials of an Universal Indicator

The proposed universal indicator MGHGINT is an indicator of production performance of different regions in terms of the GHG emissions. However, it can be used as a base to define and propose other indicators. For example, in Figure II-2, a pyramid stack of indicators are built on top of the MGHGINT. As can be seen from the figure, on top of the pyramid a carbon border tax is defined that can be used to impose penalties based on the footprint of nations calculated according to their MGHGINT indicator. However, it should be noted that the possibilities are

not just limited to the case shown in Figure II-2, and many other combinations and designs could be considered.

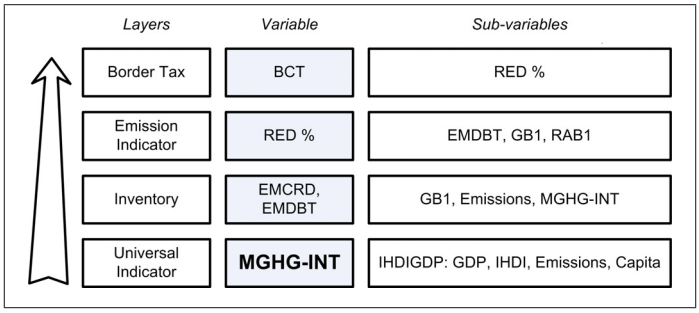


Figure-A II-2 The proposed framework which is based on a universal indicator and consists of several layers.

4 The Design of an Universal Indicator

The MGHGINT indicator is designed based on well-accepted concepts. In particular, two well-known indicators, namely the Purchasing Power Parity GDP (GDP(PPP)) and Inequality-adjusted Human Development Index (IHDI) compromise the foundation of the MGHGINT indicator. These high-level variables ensure that the calculations associated to this indicator are simple, and therefore validating the results can be easily performed by any person without facing any black box or hidden models. The MGHGINT is designed to converge two divergent aspects of nations across the globe, i.e., the population and production. This has been performed by considering "hidden" and "internal" activities associated with population that may not directly accounted for in the GDP. These internal activities are calculated based on the IHDI in order to avoid over counting a population that may implicitly promote increase in the population. Increase in population is a threat to the global sustainability and stability by itself. At the same time, the inclusion of IHDI in the definition of the MGHGINT brings the inequality aspects of nations to the picture and their performance, and therefore it would implicitly guide their policy makers to develop policies that work toward reducing inequality in the population.

5 The MGHG-INT Picture of the World in 2009

The nations MGHGINTs in 2009 are shown in Figure II-3. The normalized picture to the MGHGINT of China is also presented in order to have a better understanding with respect to big economies. It can be easily observed that all major polluters have a comparable scale of the MGHGINT, and therefore can be addressed using a same mechanism of carbon border tax or adjustment. Also, it is worth nothing that the European region is mainly green because that this study does not consider imported pollution. In future, we will use multi-region input output (MRIO) models to include the imported emissions and also have a more accurate picture of GHG footprint across the globe.

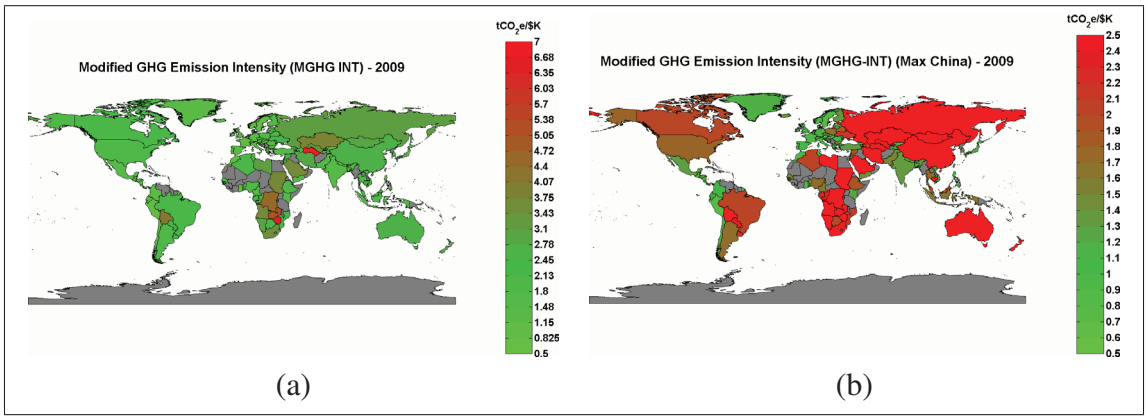


Figure-A II-3 a) The MGHGINT distribution over the world in 2009 (in $GtCO_2e/\$B$). b) The same as (a), but with the maximum value of China's MGHGINT, in order to provide a clearer picture.

6 IHDI-adjusted Gross Domestic Product (IHDIGDP) and Modified GHG Emission Intensity (MGHG-INT) Formulism

The GHG-INT of a country is defined as the ratio of its emissions to its GDP:[1]

$$\text{GHGINT}_{i,y} = \frac{\text{EM}_{i,y}}{\text{GDP}_{i,y}} \qquad \text{(A II-1)}$$

[1] We use the GDP at Purchasing Power Parity exchange rates: GDP (PPP).

where $\text{GHGINT}_{i,y}$ is the GHG-INT of country i in year y, and $\text{EM}_{i,y}$ is the total emissions of that country in the same year (excluding land-use emissions). This measure provides the GHG footprint of countries based on their *economic output*. However, countries like China with a large population prefer to use a different measure, which is a ratio of emissions to the population (GHGpCapita):

$$\text{GHGpCapita}_{i,y} = \frac{\text{EM}_{i,y}}{\text{Capita}_{i,y}} \tag{A II-2}$$

where $\text{GHGpCapita}_{i,y}$ is the GHGpCapita of country i in year y, and $\text{Capita}_{i,y}$ is its population.

To arrive at a universal GHG emissions measure which is robust with respect to variations in GDP and population, but works for all countries, we modify the GHG-INT, and redefine it as the ratio of emissions to "activities":

$$\text{MGHGINT}_{i,y} = \frac{\text{EM}_{i,y}}{\text{“activities”}_{i,y}} \tag{A II-3}$$

where $\text{MGHGINT}_{i,y}$ is the modified GHG intensity measure of country i in year y (defined above), and $\text{“activities”}_{i,y}$ is the activity of that country (explained below) during the same period. Here, "activities" replaces GDP in Equation (A II-1). We model them as an IHDI-adjusted version of GDP (IHDIGDP), which not only includes the production of a country (its GDP), but also considers the internal activity of its population.

Using the IHDIGDP, we redefine MGHG-INT as follows:

$$\text{MGHGINT}_{i,y} = \frac{\text{EM}_{i,y}}{\text{IHDIGDP}_{i,y}} \tag{A II-4}$$

where $\text{EM}_{i,y}$ represents the total GHG emissions of that country, except for the land-use CO_2 emissions.

Let us start first with IHDIxCapita. The IHDIxCapita is defined as the product of the UN Development Programme's IHDI and the population snapshot:

$$\text{IHDIxCapita}_{i,y} = \text{IHDI}_{i,y}\text{Capita}_{i,1990} \tag{A II-5}$$

where $\text{IHDI}_{i,y}$ is the IHDI of country i in year y, $\text{Capita}_{i,1990}$ is population snapshot of country i (taken in 1990), and $\text{IHDIxCapita}_{i,y}$ is the IHDIxCapita of the same country in year y. The balanced IHDIxCapita is defined as the IHDIxCapita normalized to the maximum IHDIxCapita in the same year, scaled to the maximum GDP (PPP) of the same year:

$$\text{IHDIxCapita}_{i,y}^{\text{BAL}} = \text{GDP(PPP)}_{y}^{\text{MAX}} \frac{\text{IHDIxCapita}_{i,y}}{\text{IHDIxCapita}_{y}^{\text{MAX}}} \tag{A II-6}$$

where $\text{IHDIxCapita}_{i,y}$ is the IHDIxCapita of country i in year y, and $\text{IHDIxCapita}_{y}^{\text{MAX}}$ and $\text{GDP (PPP)}_{y}^{\text{MAX}}$ are the maximum of the IHDIxCapita and the maximum of GDP (PPP) of all countries in year y respectively. The balanced GDP, $\text{GDP}_{i,y}^{\text{BAL}}$, is the ratio of the GDP (PPP) to the GDP (PPP) of the country with the IHDI of $\text{IHDIxCapita}_{y}^{\text{MAX}}$, scaled to the maximum GDP of the same year:

$$\text{GDP}_{i,y}^{\text{BAL}} = \text{GDP(PPP)}_{y}^{\text{MAX}} \frac{\text{GDP (PPP)}_{i,y}}{\text{GDP (PPP)}_{y}^{\text{IHDI}}} \tag{A II-7}$$

where $\text{GDP (PPP)}_{y}^{\text{IHDI}}$ is the GDP (PPP) of the country with the IHDI of $\text{IHDIxCapita}_{y}^{\text{MAX}}$.

With this definition, if we calculate the balanced GDP of the country with the maximum IHDIxCapita, we obtain the GDP (PPP) of the country with the maximum GDP. Also, the balanced IHDIxCapita of a country with the maximum IHDIxCapita is again the GDP (PPP) of the country with the maximum GDP. In this way, both the balanced GDP and the balanced IHDIxCapita are normalized to the same level, and therefore it is possible to average them and calculate the IHDIGDP. The IHDIGDP is defined as follows:

$$\text{IHDIGDP}_{i,y} = Z \frac{\text{GDP}_{i,y}^{\text{BAL}} + \text{IHDIxCapita}_{i,y}^{\text{BAL}}}{2} \tag{A II-8}$$

where $\text{IHDIGDP}_{i,y}$ is the IHDIGDP of country i in year y, and $\text{IHDIxCapita}_{i,y}^{\text{BAL}}$ and $\text{GDP}_{i,y}^{\text{BAL}}$ are the balanced IHDIxCapita and the balanced GDP of that country in year y respectively. The normalization parameter Z is selected is such a way that the world IHDIGDP in 1990 is equal to the world GDP (PPP) in the same year.

7 The use case of China and the USA

Figure II-4 shows a comparison of the trend in the MGHGINT and also other indicators for China and the USA along a period of two decades. It can be easily seen that both countries show an overall increasing behavior in terms of the MGHGINT indicator. Interestingly, the IHDIGDP indicator of both countries have converged to the same level in 2009 that can mainly be attributed to increase in the China's GDP. At the same time, the GHGINT indicator has an overall decreasing behavior regardless of activities. We have noticed the same behavior for other regions, and therefore we can conclude that the GHGINT is not a good direct indicator. Finally, the GHG per Capita indicator is still highly divergent between the two regions even in 2009 that prevent using this indicator as a universal indicator in policy making.

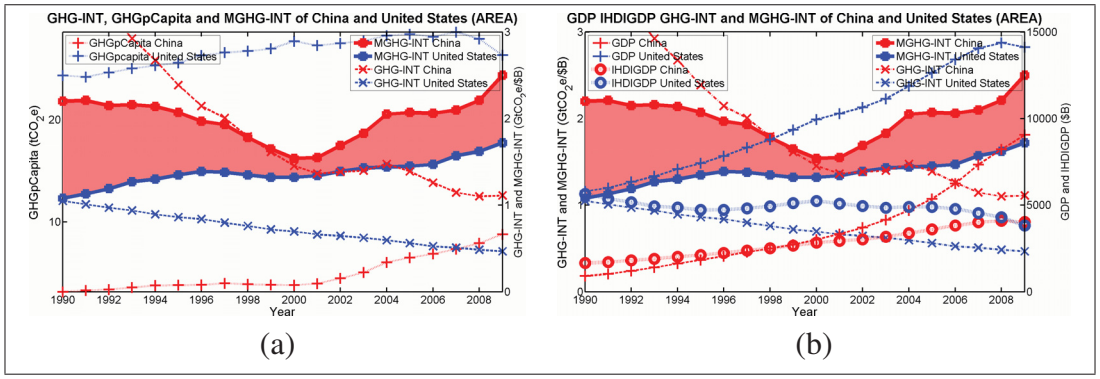


Figure-A II-4 a) A comparison between the GHG-INT, GHGpCapita, and MGHG-INT of China and the United States over two decades. b) The same comparison as in (a), but with respect to GDP, IHDIGDP, GHG-INT, and MGHG-INT.

8 Carbon Border Adjustments and two fictional tax Scenarios (2010-2020)

In order to evaluate the possible impact of the proposed carbon border tax mechanism, two fictional scenarios are considered. The first scenario is the business as usual (BAU) and is called the NC scenario. In this scenario no carbon border tax is considered. In the second scenario, called the CT scenario, a carbon border tax of one hundredth of the RED%[2] of each country is assumed starting from 2010. Figure II-5 shows both scenarios in terms of economic growth and also GHG footprint at the global scale. As can be seen, the CT scenario not only

[2]Please refer to (Farrahi Moghaddam *et al.*, 2013) for details.

helped to contain the GHG emissions at the level of 2010, it has not put a strong hurdle in front of the global economic growth.

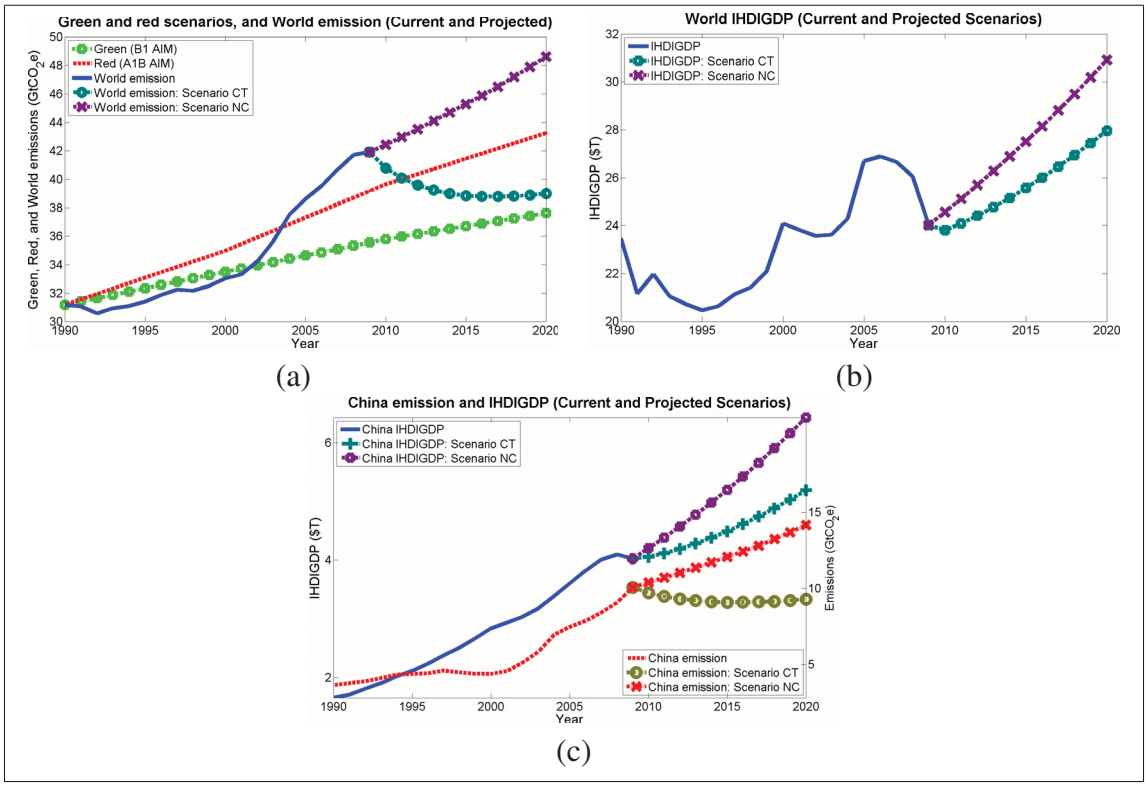


Figure-A II-5 a) The impact of the proposed BCT on global emissions in the short term. In the CT scenario, the tax is implemented, and in the NC scenario, it is business as usual. b) The impact on global economic growth. c) The impact on China's economy and emissions.

Also, Figure II-5(c) provides the China's picture with respect to NC and CT scenarios. Again, it can be easily observed that the CT scenario maintains the China's economic growth while keeping its emissions almost at the level of 2009.

9 The Next Steps

The main drawback of the proposed MGHGINT indicator is its insensitivity to the imported GHG emissions. It seems that these emissions will play a great role in future and therefore should be considered in models and indicator. We plan to generalize the MGHGINT toward an

indicator that also considers imported emissions using databases such as the GTAP databases and a multi-region input output (MRIO) model. In addition, the IHDI indicator of human development can be improved and modified in order to make it more aware of the regional variations in terms of GHG emissions; it is probable that to perform the same level of activity in two different region in the world a totally different amount of GHG emissions would be produced even if the same technology is use. This could be mainly because of differences in terms of climate and temperature. In order to avoid misuse of the IHDI indicator, an extension of this indicator is required that is emission-neural. Finally, as well known, the GHG emissions are not the only critical factors in the global sustainability. We will work to include other vital resources, such as water, energy and low temperature, in our models and indicators in order to have a better picture, and then better policies toward a sustainable world.

ANNEX III

GREENSTAR NETWORK PROJECT

GSN is a project which its aim was to create a zero carbon footprint network based on follow the sun/follow the wind methodology. The main idea behind follow the sun/follow the wind methodology is to have several interconnected data centers which are supplied with intermittent renewable sources of energy, and migrate virtual machines from locations where the green source of energy is not available at the moment to places where it is available. The resources will be brought back to this location when the renewable energy is available again. To create such a network, several components need to be designed and developed including infrastructure, middleware, and controller. Figure III-1 shows the map of GSN project. In the following section the controller of GSN project is discussed.

1 Controller

After successful experiments that confirmed that it is possible to manually live migrate (Farrahi Moghaddam and Cheriet, 2010) VMs from one location to another using the hypervisors and middleware used in the GSN system, there was a need to develop a controller that automatically performs this task toward achieving the goal of the GSN system which was zero carbon-footprint operation. However, zero carbon footprint is not achievable by 100%, since every renewable source of energy has a small portion of carbon footprint. Therefore, the goal of system is rather to minimize the carbon footprint. The controller needs to take into consideration the current active VMs and their resource usage, available servers, available renewable energy in each location, and also greenness of energy in each location. Two controllers were designed for GSN project. A greedy optimizer and a heuristic one. The greedy optimizer works based on estimation of operating hour in each location. The operating hour refers to how many hour a data center can be operational based on energy stored in the batteries. Based on this metric, locations divided to move-from and move-to groups and a greedy algorithm was used to make a plan for placement of move-from VMs on move-to servers. On the other hand, the heuristic algorithm was a primary version of MLGGA which was described in chapter 5 (Far-

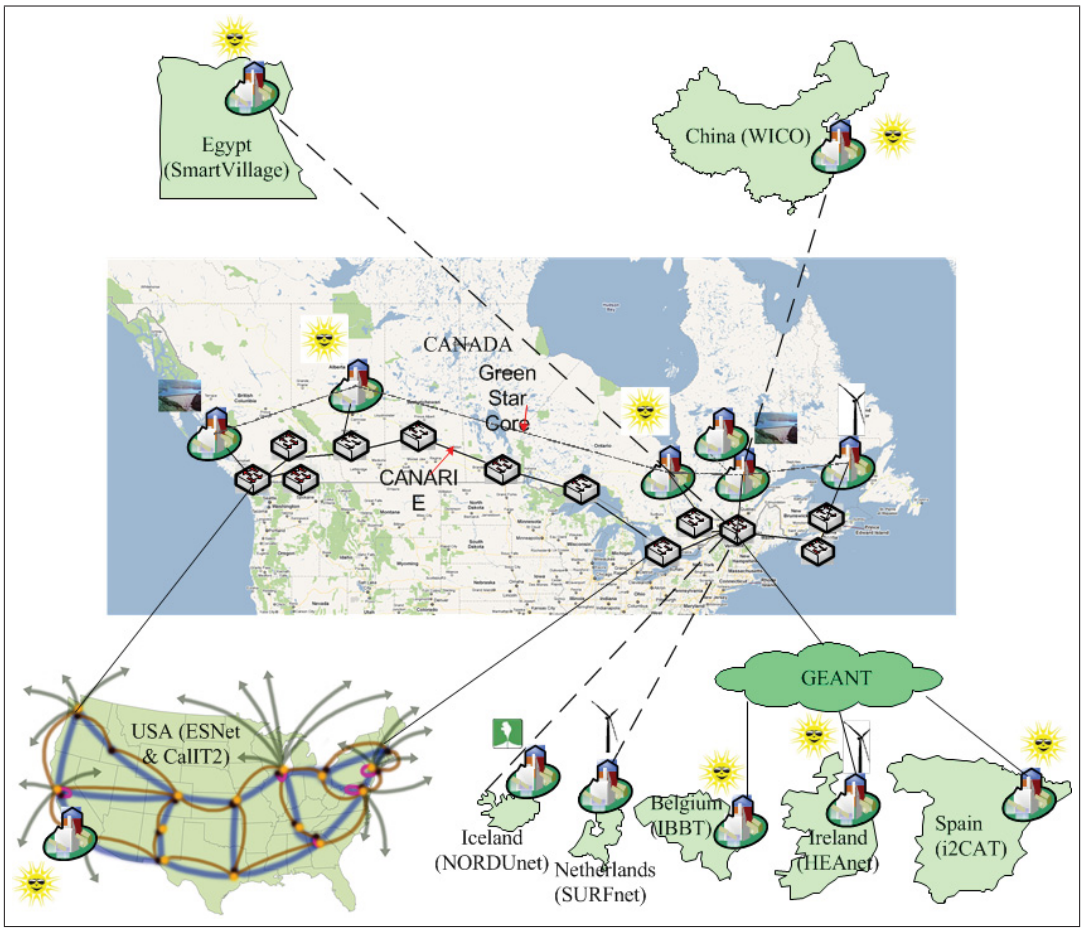


Figure-A III-1 GreenStar Network map

rahi Moghaddam *et al.*, 2012a). Result of experiments on a simulation platform reveals that in small scale networks, the greedy and heuristic algorithm has almost similar results, but in large scale networks the heuristic algorithm has better results. The heuristic algorithm is slower than the greedy algorithm in all cases.

BIBLIOGRAPHY

Abraham, Ajith, Rajkumar Buyya, and Baikunth Nath. 2000. " Nature's heuristics for scheduling jobs on computational grids ". In *ADCOM'00*. p. 45–52.

Adamou, Adamos, Sofronis Clerides, and Theodoros Zachariadis. August 2012. " Trade-offs in CO2-oriented vehicle tax reforms: A case study of Greece ". *Transportation Research Part D: Transport and Environment*, vol. 17, n° 6, p. 451–456.

Agrawal, Shubham, Sumit Kumar Bose, and Srikanth Sundarrajan. 2009. " Grouping Genetic Algorithm for Solving the Serverconsolidation Problem with Conflicts ". In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. (Shanghai, China 2009), p. 1–8. ACM.

Alkire, Sabina and James Foster. 2010. *Designing the inequality-adjusted human development index (IHDI)*. Technical Report Human Development Research Paper 2010/28. United Nations Development Programme.

Beloglazov, Anton, Rajkumar Buyya, Choon Lee, Young, and Albert Zomaya. June 30 2010. *A Taxonomy and Survey of Energy-Efficient Data centers and Cloud Computing Systems*. Technical report. University of Melbourne, Australia : CLOUDS-TR-2010-3. Cloud Computing and Distributed Systems Laboratory.

Berl, Andreas, Erol Gelenbe, Marco Di Girolamo, Giovanni Giuliani, Hermann De Meer, Minh Q. Dang, and Kostas Pentikousis. 2010. " Energy-Efficient Cloud Computing ". *The Computer Journal*, vol. 53, n° 7, p. 1045-1051.

Berral, Josep Ll., Í nigo Goiri, Ramón Nou, Ferran Julià, Jordi Guitart, Ricard Gavaldà, and Jordi Torres. 2010. " Towards energy-aware scheduling in data centers using machine learning ". In *e-Energy'10*. (Passau, Germany 2010), p. 215–224. ACM.

Bertran, R., Y. Becerra, D. Carrera, V. Beltran, M. Gonzalez, X. Martorell, J. Torres, and E. Ayguade. 2010a. " Accurate Energy Accounting for Shared Virtualized Environments Using PMC-Based Power Modeling Techniques ". In *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*. p. 1–8.

Bertran, Ramon, Marc Gonzalez, Xavier Martorell, Nacho Navarro, and Eduard Ayguade. 2010b. " Decomposable and Responsive Power Models for Multicore Processors Using Performance Counters ". In *Proceedings of the 24th ACM International Conference on Supercomputing*. (New York, NY, USA 2010), p. 147–158. ACM.

Braathen, NilsAxel. 2012. CO2-based taxation of motor vehicles. Zachariadis, T. I., editor, *Cars and Carbon*, chapter 8, p. 181–200. Springer Netherlands. doi: 10.1007/978-94-007-2123-4_8.

Braun, Tracy D, Howard Jay Siegel, Noah Beck, Ladislau L Bölöni, Muthucumaru Maheswaran, Albert I Reuther, James P Robertson, Mitchell D Theys, Bin Yao, Debra Hensgen, and Richard F Freund. June 2001. " A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems ". *Journal of Parallel and Distributed Computing*, vol. 61, n° 6, p. 810–837.

Buyya, R., R. Ranjan, and R.N. Calheiros. 2009. " Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities ". In *HPCS '09*. p. 1–11.

Caron, E., F. Desprez, and A. Muresan. Nov 30-Dec 3 2010. " Forecasting for Grid and Cloud Computing On-Demand Resources Based on Pattern Matching ". In *CloudCom'10*. (Indianapolis, IN, USA 2010), p. 456–463.

Chen, Shiyi. September 2013. " What is the potential impact of a taxation system reform on carbon abatement and industrial growth in China? ". *Economic Systems*, vol. 37, n° 3, p. 369–386.

Chen, Yiyu, Amitayu Das, Wubi Qin, Anand Sivasubramaniam, Qian Wang, and Natarajan Gautam. 2005. " Managing server energy and operational costs in hosting centers ". In *ACM SIGMETRICS Performance Evaluation Review*. p. 303–314. ACM.

Chua, Seng Tat and Masaru Nakano. 2013. Design of a taxation system to promote electric vehicles in Singapore. Emmanouilidis, C., Marco Taisch, and Dimitris Kiritsis, editors, *IFIP Advances in Information and Communication Technology*, volume 397, p. 359–367. Springer. doi: 10.1007/978-3-642-40352-1_45.

CORNWELL, ANTONIA and JOHN CREEDY. August 1996. " Carbon Taxation, Prices and Inequality in Australia ". *Fiscal Studies*, vol. 17, n° 3, p. 21–38.

Das, Rajarshi, Jeffrey O Kephart, Jonathan Lenchner, and Hendrik Hamann. July 7–11 2010. " Utility-function-driven energy-efficient cooling in data centers ". In *ICAC'10*. (Washington, DC, USA 2010), p. 61–70. ACM.

de Assuncao, Marcos Dias, Alexandre di Costanzo, and Rajkumar Buyya. JUne 11-13 2009. " Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters ". In *HPDC'09*. (Garching, Germany 2009), p. 141–150. ACM.

Energy Design Resources. June 2010. " Design Brief: Chiller Plant Efficiency ". <http://www.energydesignresources.com/resources/publications/design-briefs/design-brief-chiller-plant-efficiency.aspx>. Latest accessed on April 20th, 2013.

Etinski, Maja, Julita Corbalan, Jesus Labarta, and Mateo Valero. 2010. " Utilization driven power-aware parallel job scheduling ". *Computer Science - Research and Development*, vol. 25, n° 3-4, p. 207–216.

F. Julià,J. Roldàn, R. Nou O. Fitó Vaquè Í Goiri J. Berral. 2010. *EEFSim: energy efficency simulator*. Technical Report UPC-DAC-RR-2010-19. Spain : Universitat Politècnica de Catalunya.

Falkenauer, E. and A. Delchambre. 1992. “A Genetic Algorithm for Bin Packing and Line Balancing”. In *IEEE International Conference on Robotics and Automation*. p. 1186–1192 vol.2.

Farrahi Moghaddam, F. and M. Cheriet. 2010. “Decreasing Live Virtual Machine Migration Down-Time Using a Memory Page Selection Based on Memory Change PDF”. In *Networking, Sensing and Control (ICNSC), 2010 International Conference on*. p. 355–359.

Farrahi Moghaddam, Fereydoun, M. Cheriet, and Kim Khoa Nguyen. July 4-9 2011. “Low Carbon Virtual Private Clouds”. In *IEEE International Conference on Cloud Computing (CLOUD’ 11)*. (Washington, DC, USA 2011), p. 259–266.

Farrahi Moghaddam, Fereydoun, Reza Farrahi Moghaddam, and Mohamed Cheriet. April 18-21 2012a. “Multi-Level Grouping Genetic Algorithm for Low Carbon Virtual Private Clouds”. In *2nd International Conference on Cloud Computing and Services Science (CLOSER’12)*. (Porto, Portugal 2012), p. 315–324.

Farrahi Moghaddam, Fereydoun, Reza Farrahi Moghaddam, and Mohamed Cheriet. June 2012b. “Carbon Metering and Effective Tax Cost Modeling for Virtual Machines”. In *IEEE Fifth International Conference on Cloud Computing*. (Honolulu, Hawaii, USA 2012), p. 758-763.

Farrahi Moghaddam, Reza, Fereydoun Farrahi Moghaddam, and Mohamed Cheriet. 2013. “A modified GHG intensity indicator: Toward a sustainable global economy based on a carbon border tax and emissions trading”. *Energy Policy*, vol. 57, n° 0, p. 363–380.

Feng, Wu-chun, Xizhou Feng, and Rong Ce. 2008. “Green Supercomputing Comes of Age”. *IT Professional*, vol. 10, n° 1, p. 17–23.

Freund, R.F., Michael Gherrity, S. Ambrosius, M. Campbell, M. Halderman, D. Hensgen, E. Keith, T. Kidd, M. Kussow, J.D. Lima, F. Mirabile, L. Moore, B. Rust, and H.J. Siegel. 1998. “Scheduling resources in multi-user, heterogeneous, computing environments with SmartNet”. In *HCW’98*. p. 184–199.

Fumo, Nelson, Pedro J. Mago, and Kenneth Jacobs. February 2011. “Design considerations for combined cooling, heating, and power systems at altitude”. *Energy Conversion and Management*, vol. 52, n° 2, p. 1459–1469.

Gagoa, Alberto, Xavier Labandeira, and Xiral López-Otero. 2013. “A Panorama on Energy Taxes and Green Tax Reforms”. *Ecomonics for Energy*, vol. WP 08/2013, p. 1-45.

Gandhi, Anshul, Mor Harchol-Balter, Rajarshi Das, and Charles Lefurgy. 2009. “Optimal power allocation in server farms”. *SIGMETRICS Perform. Eval. Rev.*, vol. 37, n° 1, p. 157–168.

Garey, Michael R. and David S. Johnson. 1979. *A Guide to The Theory of NP-Completeness*. Technical report. San Francisco : W.H.Freeman Co.

Garg, Saurabh Kumar, Chee Shin Yeo, Arun Anandasivam, and Rajkumar Buyya. June 2011. "Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers". *Journal of Parallel and Distributed Computing*, vol. 71, n° 6, p. 732–749.

Gemechu, Eskinder Demisse, Isabela Butnar, Maria Llop, and Francesc Castells. May 2013. "Economic and environmental effects of CO2 taxation: an input-output analysis for Spain". *Journal of Environmental Planning and Management*, vol. Online First, p. 1–18.

GeSI. 2008. "Smart 2020: Enabling the Low Carbon Economy in the Information Age". <smart2020.org/_assets/files/02_Smart2020Report.pdf>.

Global Commerce Initiative and Capgemini. 2008. "Future supply chain 2016". *See http://www.capgemini.com/insights-and-resources/bypublication/future_supply_chain_2016*.

Goiri, Íñigo, Josep Ll Berral, J Oriol Fitó, Ferran Julià, Ramon Nou, Jordi Guitart, Ricard Gavaldà, and Jordi Torres. 2012. "Energy-efficient and multifaceted resource management for profit-driven virtualized data centers". *Future Generation Computer Systems*, vol. 28, n° 5, p. 718–731.

Gupta, R., S.K. Bose, S. Sundarrajan, M. Chebiyam, and A. Chakrabarti. july 2008. "A Two Stage Heuristic Algorithm for Solving the Server Consolidation Problem with Item-Item and Bin-Item Incompatibility Constraints". In *Services Computing, 2008. SCC '08. IEEE International Conference on*. p. 39 -46.

Guzek, Mateusz, CesarO. Diaz, JohnatanE. Pecero, Pascal Bouvry, and AlbertY. Zomaya. 2012. Impact of voltage levels number for energy-aware bi-objective DAG scheduling for multi-processors systems. Papasratorn, B., Nipon Charoenkitkarn, Kittichai Lavangnananda, Wichian Chutimaskul, and Vajirasak Vanijja, editors, *Communications in Computer and Information Science*, volume 344, p. 70-80–. Springer Berlin Heidelberg. doi: 10.1007/978-3-642-35076-4_7.

Haas, J, JAMIE Froedge, J Pflueger, and D Azevedo. 2009. *Usage and public reporting guidelines for the green grid's infrastructure metrics (PUE/DCiE)*. The Green Grid's White Paper 22. The Green Grid.

Hong, Angela C., Cora J. Young, Michael D. Hurley, Timothy J. Wallington, and Scott A. Mabury. 2013. "Perfluorotributylamine: A novel long-lived greenhouse gas". *Geophys. Res. Lett.*, vol. 40, n° 22, p. 6010–6015.

Hwang, Jinho, Sai Zeng, Timothy Wood, et al. 2013. "Benefits and challenges of managing heterogeneous data centers". In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*. p. 1060–1065. IEEE.

Iosup, A. and D. Epema. 2011. "Grid Computing Workloads". *IEEE Internet Computing*, vol. 15, n° 2, p. 19–26.

Jekabsons, G. 2011. “ARESLab: Adaptive Regression Splines toolbox for Matlab/Octave”. available at http://www.cs.rtu.lv/jekabsons/ [accessed on Oct 25th, 2012].

Jotzo, Frank and John Pezzey. June 2005. *Optimal intensity targets for emissions trading under uncertainty*. Economics and Environment Network Working Paper EEN0504. Australian National University.

Kansal, Aman, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A. Bhattacharya. 2010. “Virtual Machine Power Metering and Provisioning”. In *Proceedings of the 1st ACM symposium on Cloud computing*. (Indianapolis, Indiana, USA 2010), p. 39–50. ACM.

Kessaci, Y., N. Melab, and E. Talbi. 2011. “A pareto-based GA for scheduling HPC applications on distributed cloud infrastructures”. In *HPCS’11*. p. 456–462.

Kim, Jong-Kook, S. Shivle, H.J. Siegel, A.A. Maciejewski, T.D. Braun, M. Schneider, S. Tideman, R. Chitta, R.B. Dilmaghani, R. Joshi, A. Kaul, A. Sharma, S. Sripada, P. Vangari, and S.S. Yellampalli. 22-26 April 2003. “Dynamic mapping in a heterogeneous environment with tasks having priorities and multiple deadlines”. In *IPDPS’03*. (Nice, France 2003), p. 15 pp.

Kliazovich, Dzmitry, Pascal Bouvry, and SameeUllah Khan. 2012. “GreenCloud: a packet-level simulator of energy-aware cloud computing data centers”. *The Journal of Supercomputing*, vol. 62, n° 3, p. 1263–1283.

Kołodziej, Joanna, SameeUllah Khan, Lizhe Wang, Aleksander Byrski, Nasro Min-Allah, and SajjadAhmad Madani. August 2012. “Hierarchical genetic-based grid scheduling with energy optimization”. *Cluster Computing*, p. 1–19.

Laurent, Alexis, Stig I. Olsen, and Michael Z. Hauschild. March 2012. “Limitations of Carbon Footprint as Indicator of Environmental Sustainability”. *Environ. Sci. Technol.*, vol. 46, n° 7, p. 4100–4108.

Lawson, Barry and Evgenia Smirni. 2005. “Power-aware resource allocation in high-end systems via online simulation”. In *ICS’05*. (Cambridge, Massachusetts, USA 2005), p. 229–238. ACM.

Le, Kien, R. Bianchini, T.D. Nguyen, O. Bilgir, and M. Martonosi. August 15-18 2010. “Capping the brown energy consumption of Internet services at low cost”. In *Green Computing Conference, 2010 International*. (Chicago, IL, USA 2010), p. 3–14.

Lee, W.L. and S.H. Lee. March 2007. “Developing a simplified model for evaluating chiller-system configurations”. *Applied Energy*, vol. 84, n° 3, p. 290–306.

Lenzen, Manfred. 2010. “Current State of Development of Electricity-Generating Technologies: A Literature Review”. *Energies*, vol. 3, p. 462-591.

Lim, Seung-Hwan, B. Sharma, Gunwoo Nam, Eun Kyoung Kim, and C.R. Das. 2009. “MDCSim: A multi-tier data center simulation, platform”. In *CLUSTER ’09*. p. 1–9.

Lingrand, Diane, Johan Montagnat, Janusz Martyniak, and David Colling. 2010. "Optimization of Jobs Submission on the EGEE Production Grid: Modeling Faults Using Workload". *Journal of Grid Computing*, vol. 8, n° 2, p. 305–321.

Liu, Liang, Hao Wang, Xue Liu, Xing Jin, Wen Bo He, Qing Bo Wang, and Ying Chen. 2009. "GreenCloud: A New Architecture for Green Data Center". In *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*. (Barcelona, Spain 2009), p. 29–38. ACM.

Liu, Zhenhua, Minghong Lin, Adam Wierman, Steven H Low, and Lachlan LH Andrew. 2011. "Greening geographical load balancing". In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. p. 233–244. ACM.

Lundgren, Tommy and Per-Olov Marklund. February 17 2012. "Environmental Performance and Profits". *CERE Working Paper*, , p. 1–18.

Maheswaran, Muthucumaru, Shoukat Ali, Howard Jay Siegel, Debra Hensgen, and Richard F. Freund. November 1999. "Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems". *Journal of Parallel and Distributed Computing*, vol. 59, n° 2, p. 107–131.

Marzolla, M., O. Babaoglu, and F. Panzieri. 2011. "Server consolidation in Clouds through gossiping". In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*. p. 1 -6.

McKinsey. November 2007. *The Impact of ICT on Global Emissions*. Technical report. tech. rep., on behalf of the Global eSustainability Initiative (GeSI).

Medina, Pamela Elena Gardea. June 2013. "Climate change mitigation, a carbon tax or an emissions trading scheme?: an analysis of the Norwegian perspective". Master's thesis, NHH - the Norwegian School of Economics, Bergen, Norway.

Molfetas, Angelos, Fernando Barreiro Megino, Andrii Tykhonov, Mario Lassnig, Vincent Garonne, Martin Barisits, Simone Campana, Gancho Dimitrov, Stephane Jezequel, Ikuo Ueda, and Florbela Tique Aires Viegas. 2011. "Popularity framework to process dataset traces and its application on dynamic replica reduction in the ATLAS experiment". *Journal of Physics: Conference Series*, vol. 331, n° 6, p. 062018(1-6).

NERA Economic Consulting. Februrary 26 2013. *Economic outcomes of a U.S. carbon tax*. Technical report. Washington, DC, USA : National Association of Manufacturers.

Nesmachnow, Sergio, Bernabé Dorronsoro, JohnatanE. Pecero, and Pascal Bouvry. May 2013. "Energy-Aware Scheduling on Multicore Heterogeneous Grid Computing Systems". *Journal of Grid Computing*, vol. Online First.

Núñez, Alberto, Jose L. Vázquez-Poletti, Agustin C. Caminero, Gabriel G. Castañé, Jesus Carretero, and Ignacio M. Llorente. 2012. "iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator". *Journal of Grid Computing*, vol. 10, n° 1, p. 185–209.

Patel, Chandrakant D, Ratnesh K Sharma, Cullen E Bash, and Monem Beitelmal. March 2006. "Energy flow in the information technology stack: coefficient of performance of the ensemble and its impact on the total cost of ownership". *HP Labs External Technical Report, HPL-2006-55*.

Pereira, Alfredo Marvão and Rui M. Pereira. May 2013. "Government behavior, endogenous growth and the economic and budgetary impact of CO2 taxation in Portugal". *Working Paper Number 105, College of William and Mary, Department of Economics*, p. 1–23.

Petrucci, Vinicius, Orlando Loques, and Daniel Moss. 2009. "A Dynamic Configuration Model for Power-Efficient Virtualized Server Clusters". In *11th Brazilian Workshop on Real-Time and Embedded Systems*.

Pop, Cristina Bianca, Ionut Anghel, Tudor Cioara, Ioan Salomie, and Iulia Vartic. 2012. "A swarm-inspired data center consolidation methodology". In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*. (New York, NY, USA 2012), p. 41:1–41:7.

Qureshi, Asfandyar, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. 2009. "Cutting the electric bill for internet-scale systems". *SIGCOMM Comput. Commun. Rev.*, vol. 39, n° 4, p. 123–134.

Rao, Lei, Xue Liu, Le Xie, and Wenyu Liu. 2010. "Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment". In *INFOCOM, 2010 Proceedings IEEE*. p. 1–9. IEEE.

Rizvandi, Nikzad Babaii, Javid Taheri, Albert Y Zomaya, and Young Choon Lee. 2010. "Linear combinations of dvfs-enabled processor frequencies to modify the energy-aware scheduling algorithms". In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*. p. 388–397. IEEE.

Rodero, I., J. Jaramillo, A. Quiroz, M. Parashar, F. Guim, and S. Poole. Aug 15-18 2010. "Energy-efficient application-aware online provisioning for virtualized clouds and data centers". In *2010 International Green Computing Conference*. (Chicago, IL, USA 2010), p. 31–45.

Sankaranarayanan, Ananth Narayan, Somsubhra Sharangi, and Alexandra Fedorova. 2011. "Global cost diversity aware dispatch algorithm for heterogeneous data centers". In *ACM SIGSOFT Software Engineering Notes*. p. 289–294. ACM.

Sawyer, Richard. 2004. *Calculating total power requirements for data centers*. White Paper 3. American Power Conversion.

Speitkamp, B. and M. Bichler. 2010. "A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers". *Services Computing, IEEE Transactions on*, vol. 3, p. 266 -278.

Srikantaiah, Shekhar, Aman Kansal, and Feng Zhao. 2008. " Energy Aware Consolidation for Cloud Computing ". In *Proceedings of the 2008 conference on Power aware computing and systems*. (San Diego, California 2008), p. 10–10. USENIX Association.

The World Bank Group. 2011. " World Development Indicators Database ". http://publications.worldbank.org/WDI/indicators, [Accessed on August 24, 2011].

Tipley, Roger. 2012. *PUE: A comprehensive examination of the metric*. The Green Grid's White Paper 49. The Green Grid, – p.

Toporkov, Victor, Anna Toporkova, Alexander Bobchenkov, and Dmitry Yemelyanov. 2011. " Resource Selection Algorithms for Economic Scheduling in Distributed Systems ". *Procedia Computer Science*, vol. 4, n° 0, p. 2267–2276.

Van der Merwe, J., K. K. Ramakrishnan, M. Fairchild, A. Flavel, J. Houle, H. A. Lagar-Cavilla, and J. Mulligan. May 5-7 2010. " Towards a ubiquitous cloud computing infrastructure ". In *17th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*. p. 1-6.

Venugopal, S., Xingchen Chu, and R. Buyya. 2008. " A Negotiation Mechanism for Advance Resource Reservations Using the Alternate Offers Protocol ". In *IWQoS'08*. p. 40–49.

Wang, Leping and Ying Lu. 2008. " Efficient power management of heterogeneous soft real-time clusters ". In *Real-Time Systems Symposium, 2008*. p. 323–332. IEEE.

Wang, Lu and U. Neumann. 20-25 June 2009. " A robust approach for automatic registration of aerial images with untextured aerial LiDAR data ". In *CVP'09*. (Miami, FL, USA 2009), p. 2623–2630.

Wang, Min and Rong Chu. 2009. " A novel white blood cell detection method based on boundary support vectors ". In *SMC'09*. p. 2595–2598.

Wilcox, D., A. McNabb, and K. Seppi. 2011. " Solving Virtual Machine Packing with A Reordering Grouping Genetic Algorithm ". In *Evolutionary Computation (CEC), 2011 IEEE Congress on*. p. 362–369.

Wood, T., K. Ramakrishnan, J. van der Merwe, and P. Shenoy. January 2010. *CloudNet: A Platform for Optimized WAN Migration of Virtual Machines*. Technical report. University of Massachusetts Technical Report TR-2010-002.

Wright, David. 2012. " Evolution of Standards for Smart Grid Communications ". *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, vol. 4, n° 1, p. 47–55.

Wright, D.J. Summer 2013. " Taming our virtual smokestacks ". *, Research Perspectives*, vol. 15, n° 1, p. 16–17.

Wu, Yongwei, Kai Hwang, Yulai Yuan, and Weimin Zheng. 2010. " Adaptive Workload Prediction of Grid Performance in Confidence Windows ". *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, n° 7, p. 925–938.

Wu, Zhangjun, Xiao Liu, Zhiwei Ni, Dong Yuan, and Yun Yang. 2013. “A market-oriented hierarchical scheduling strategy in cloud workflow systems”. *The Journal of Supercomputing*, vol. 63, n° 1, p. 256–293.

Xhafa, Fatos and Ajith Abraham. April 2010. “Computational models and heuristic methods for Grid scheduling problems”. *Future Generation Computer Systems*, vol. 26, n° 4, p. 608–621.

Xianqiang, Mao, Yang Shuqian, and Liu Qin. March 2013. *The way to CO2 emission reduction and the co-benefits of local air pollution control in China's transportation sector: A policy and economic analysis*. Technical Report rr2013036. Laguna, Philippines : Economy and Environment Program for Southeast Asia (EEPSEA), – p.

Xu, J. and J.A.B. Fortes. December 2010. “Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments”. In *In proceedings of the 2010 IEEE/ACM Inter. Conference on Green Computing and Communications & Inter. Conference on Cyber, Physical and Social Computing*. (Hangshou, PR of China 2010).

Zhang, Luna Mingyi, Keqin Li, and Yan-Qing Zhang. 2010. “Green task scheduling algorithms with speeds optimization on heterogeneous cloud servers”. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*. p. 76–80. IEEE Computer Society.

Zhang, Qi, Lu Cheng, and Raouf Boutaba. 2008. “Cloud Computing: State-of-The-Art and Research Challenges”. *Journal of Internet Services and Applications*, vol. 1, n° 1, p. 7-18.

Zimmermannová, Jarmila. 2013. “Current and Proposed CO2 Taxation in the European Union Member States in the Energy Sector”. *Acta Oeconomica Pragensia*, vol. 2013, n° 2, p. 40–54.